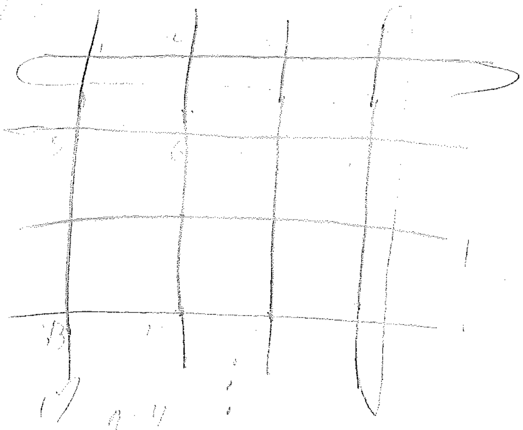
Brewster Kahle's Book

NE 43 - 3rd Floor AI LSE

check pend. clk-1-polygatk)) for



(load "AI:MMCM;salvage)
(salvage-editor)
"ai:LMFS;FS_DOC"

Key to 36-328
97513

To lisp m room

Andrew Love x5-6488

AR9: setnet;

pocket dict m1: dict;

filename=m1wpd...

Keywords for pocket dict

on m1: WBA;
; smaller dictionary

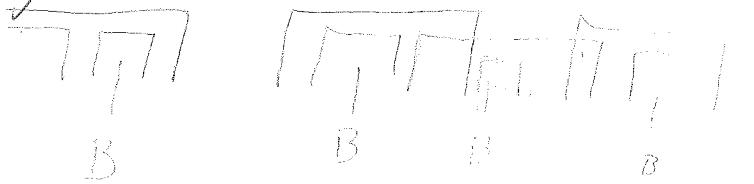
real dict: (huge)

ml: dict; mycel >

8	2	4	11
13	8	6	15
1	10	12	3
5	16	14	7

14

To get CIF out to Mosis

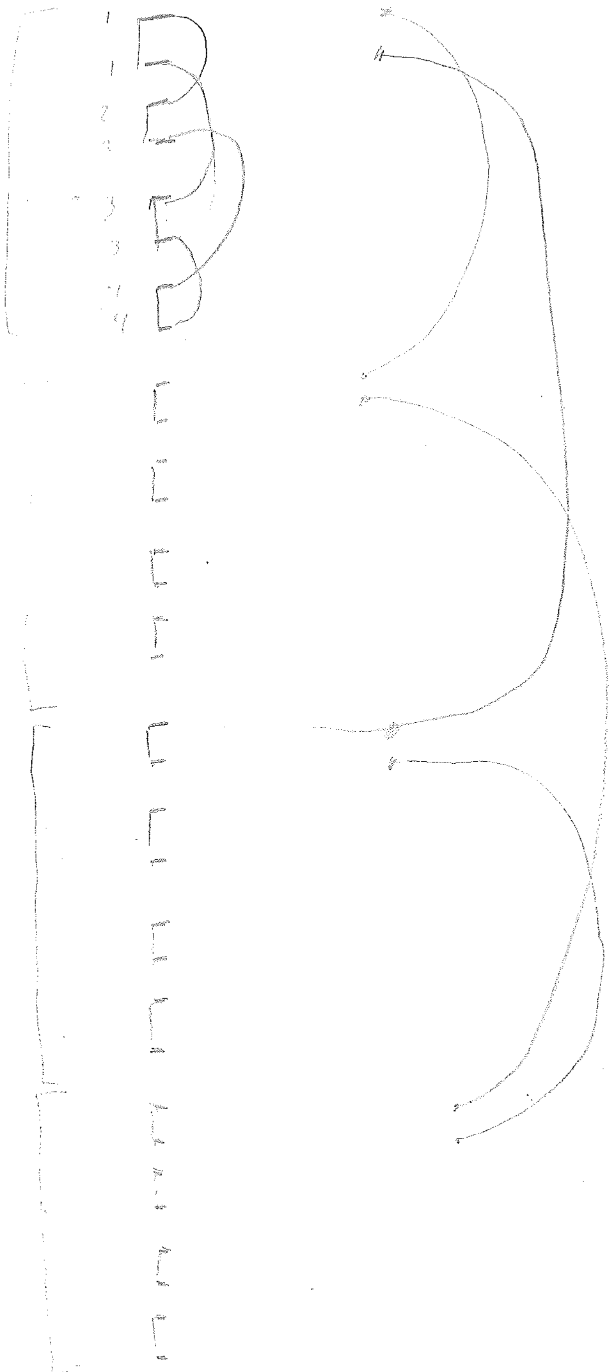


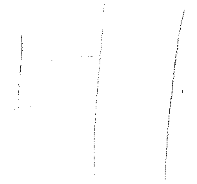
4	2	7	10
8	1	6	12
12	3	9	1
16	5		11

N1	0
S1	3
N2	0
S2	3
E1	0
W1	12
E2	0
W2	12
N1	1
S1	0
N2	1
S2	0

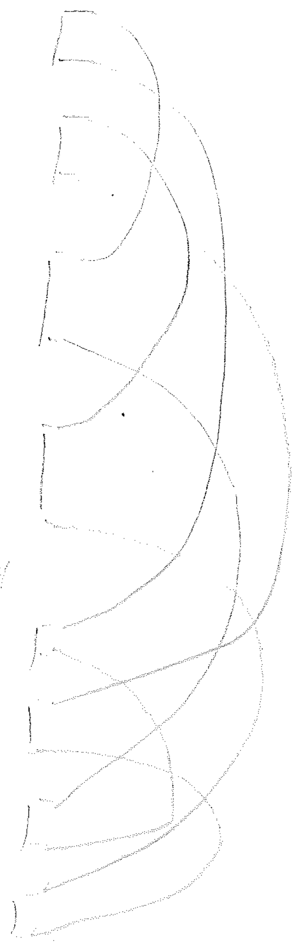
ml:WBA;spel

Phase1 Phase 2





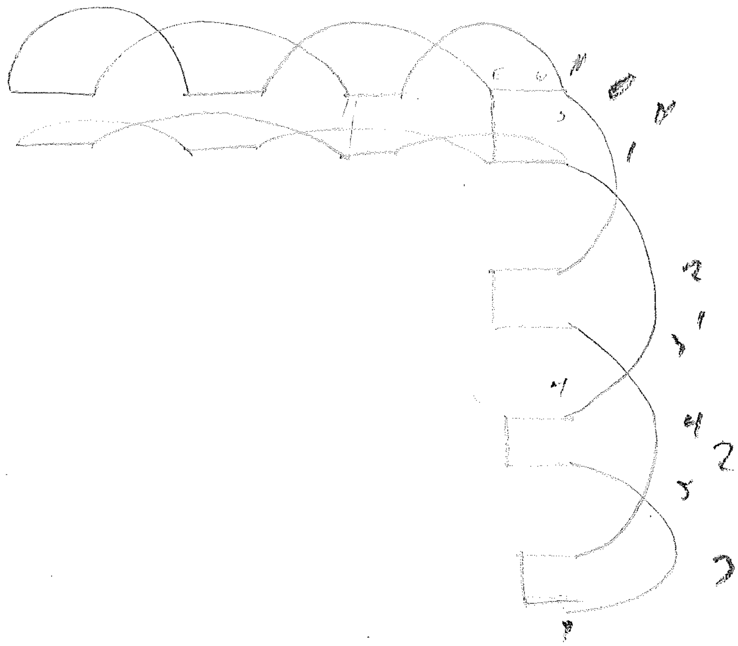
Pgth 4



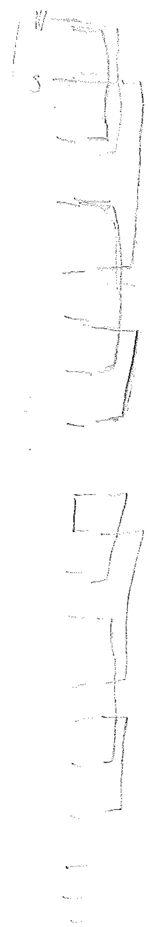
0123 → 16

01 → 4 → 16
23 → 4

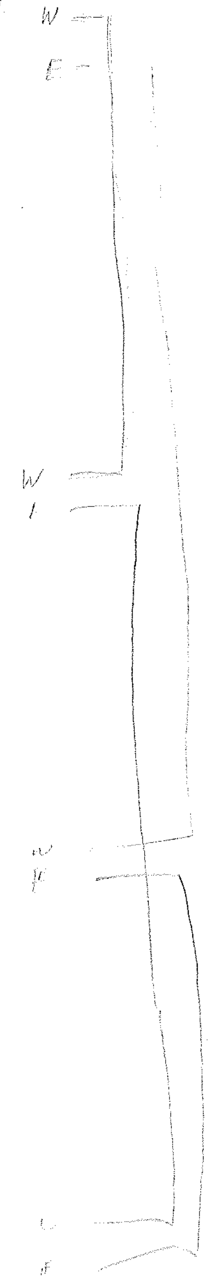
EW = 2 Each thick x 2 deep x 4 repeats
NS = 1 Each thi...



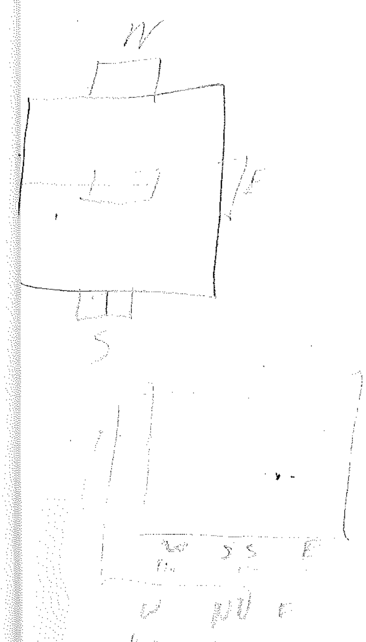
N-S



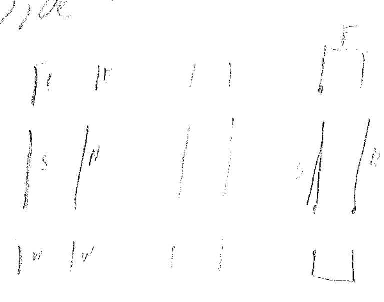
E W

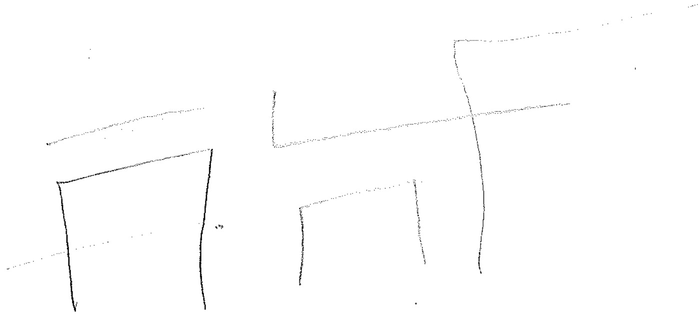


5	13	9	1
7	15	11	3
6	14	10	2
4	12	8	0

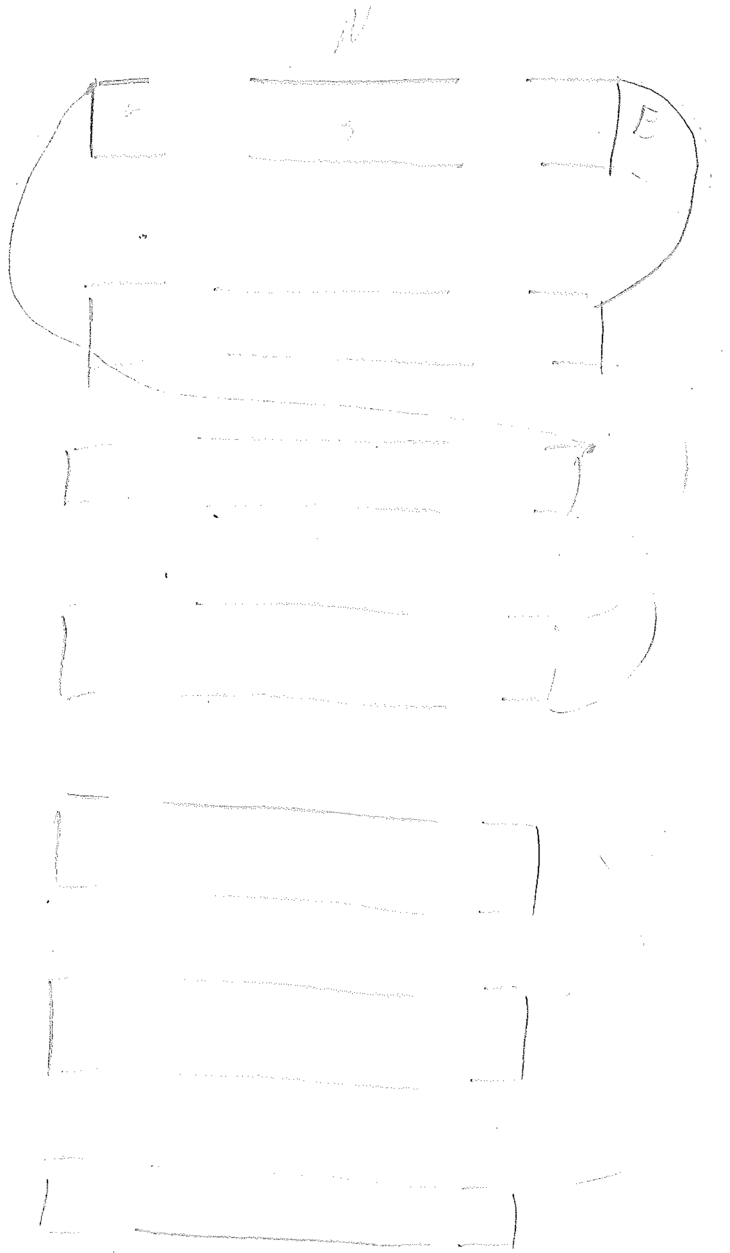


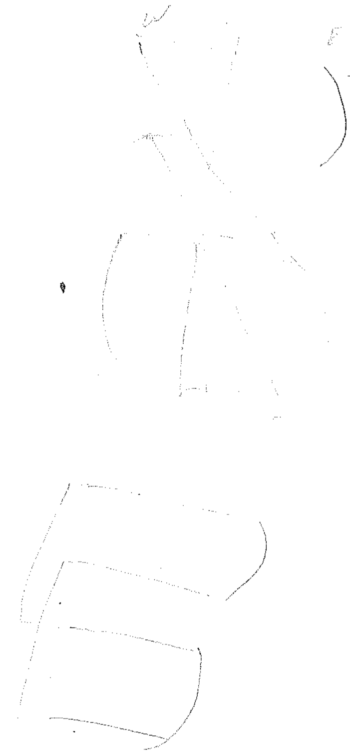
Side view



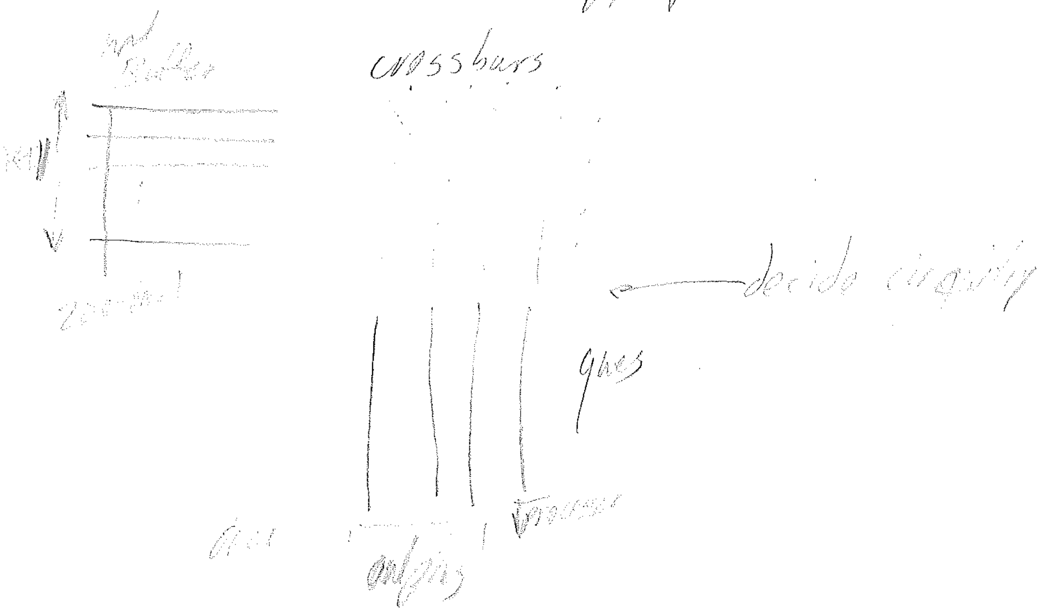


14
2 13
1 36
7 32
3 22
10 21
9
1544


N
E



simplist for total message package
drop words



How many processors?
∴ how large address
smart or dumb how secure is it?
what is message length
smart: problems
heads come in message left behind and free path to 50

group for heads only

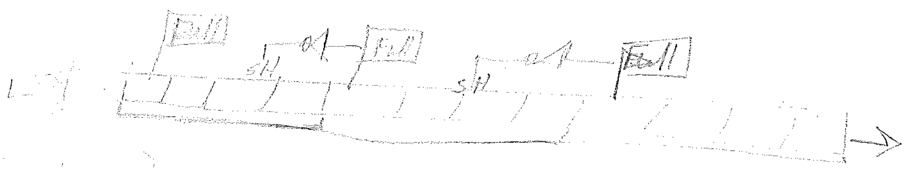
for Snakes
No Speed Brakes

Decision prob
based on destination
queue state
history of message — where it is coming from
Acts on queue

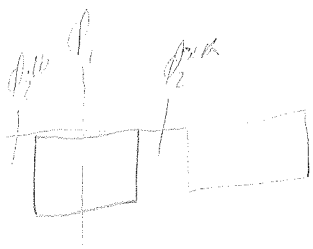
max depth
head queue
with $\log_2(k+1)$

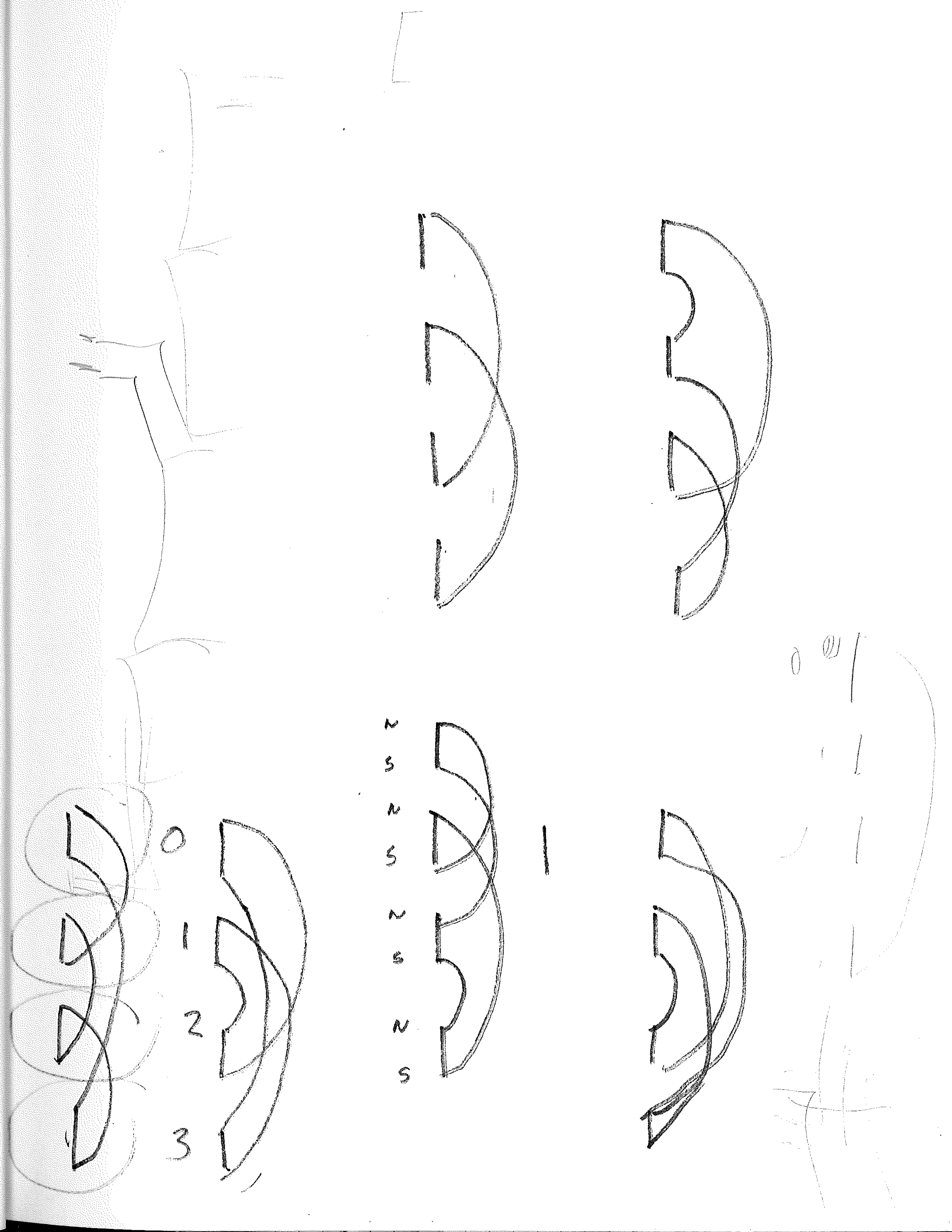
head of queue
gets acknowledge
sends which dir wants to go where

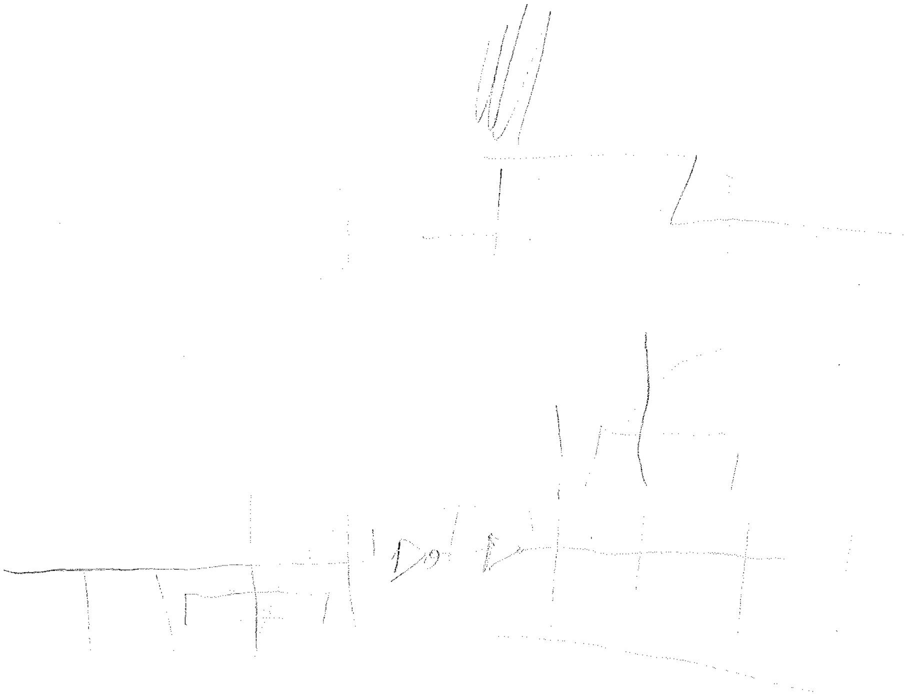
poss problem: encircling

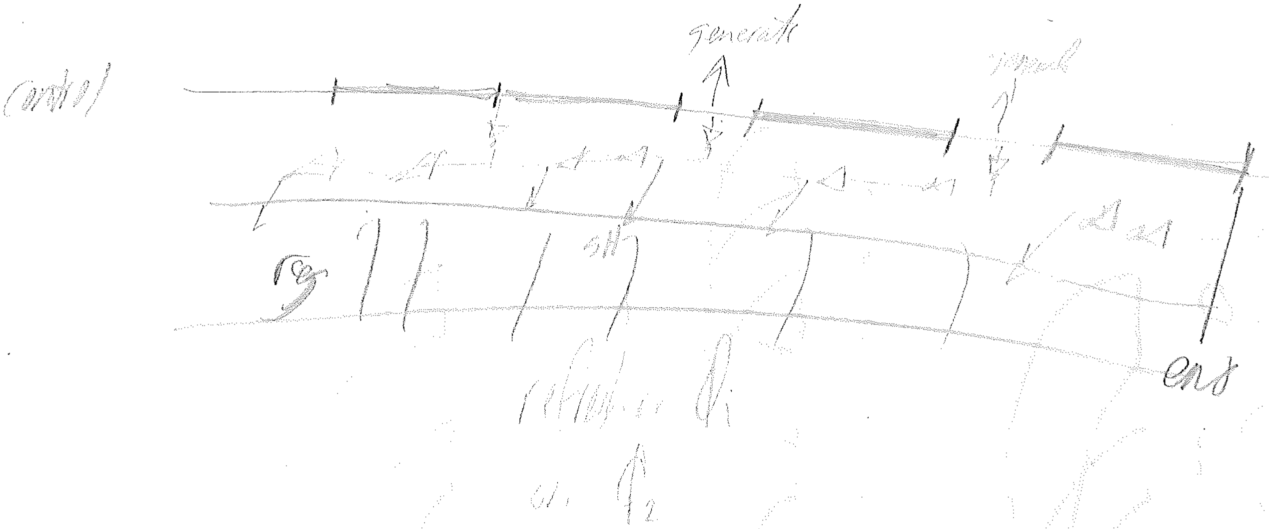


Do-Do




0
1
2
3
N
S
N
S
N
S
N
S
1



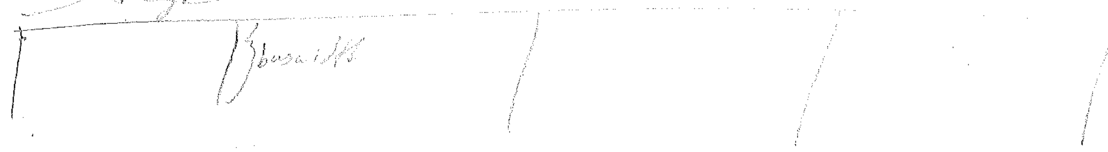

Generate
end

9MC

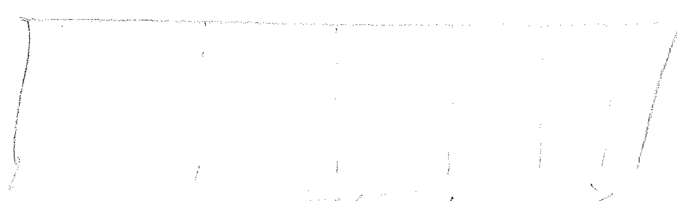
input: packages
output: gene state: Full
: new full

packages

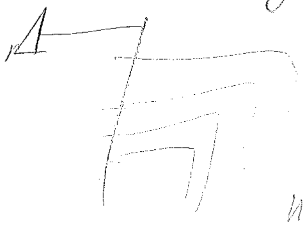
Package lengths



message length words

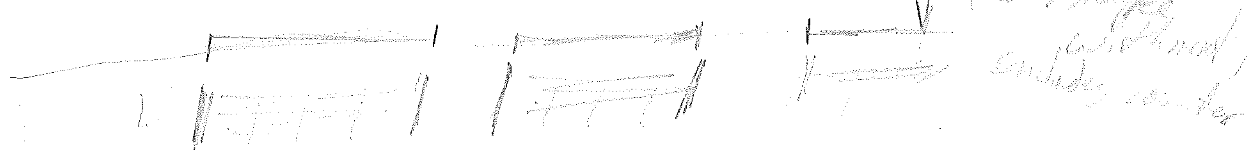


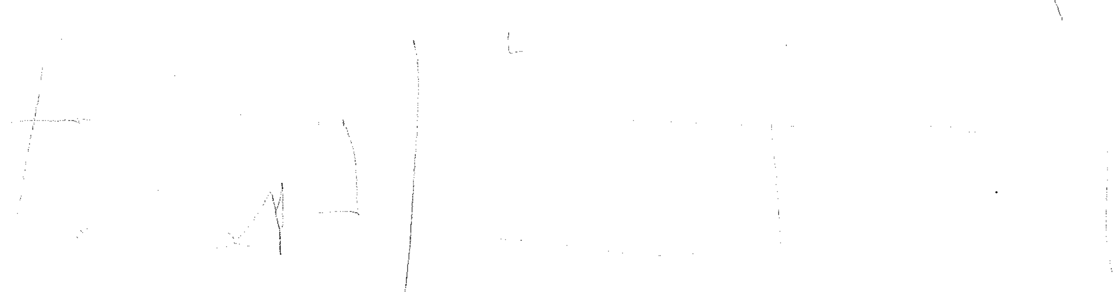
Flag for full or ~~filling~~ or emptying
set by shifting from previous
erased by finished shifting to next
used up



control for [illegible]

controling for





general

(20, 20) bit Address   Message ~ 30-100 bits long

(x, y).   # of connections ~ 5-7

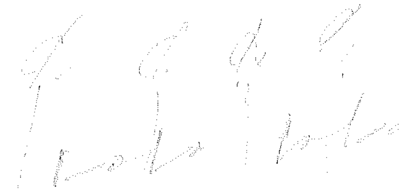
bus will be ~ 2-3 b wide

must put ~~some~~ one of it together

Address [x-field | nx-field]

on input zero detectors

reset before a message comes in
count # of zero bytes if ever ≠ 0 don't count
if it gets to the magic # then set cross bars

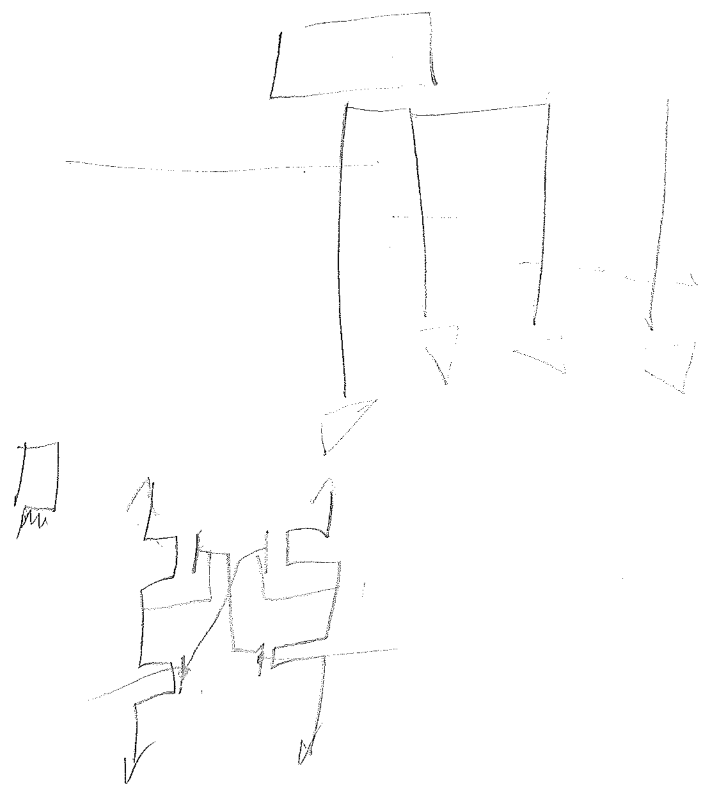
Input reg: shifting cells 

decision block: ques, # of zeros?
destination: can it tell = advantageous direction
fixed ... ?

K field 1st bit K-field

use for mask ... que states, then take smallest
que state if that is full ~~then~~ (or any ... set)
Then must go for disadvantageous direction
invert mask and retest,
if still full put in 1st free Que [= lossage]
(how to deal with ... , ...

comparator analoge?



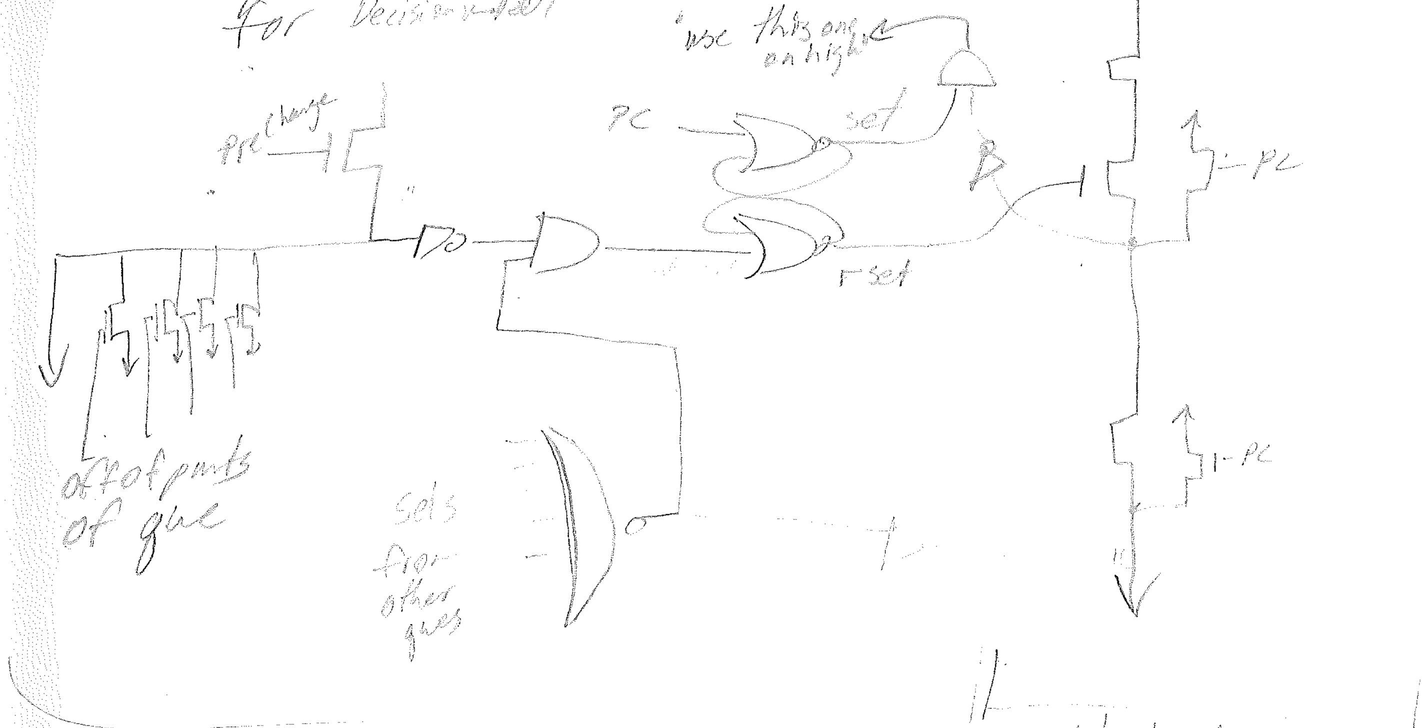
hand shake arbitration

Samples every 3

1 or 2 times

comparater by Knight

for Decisionmaker



arbitrator

phase 1

now will have some flip flops set (hopefully just a few)

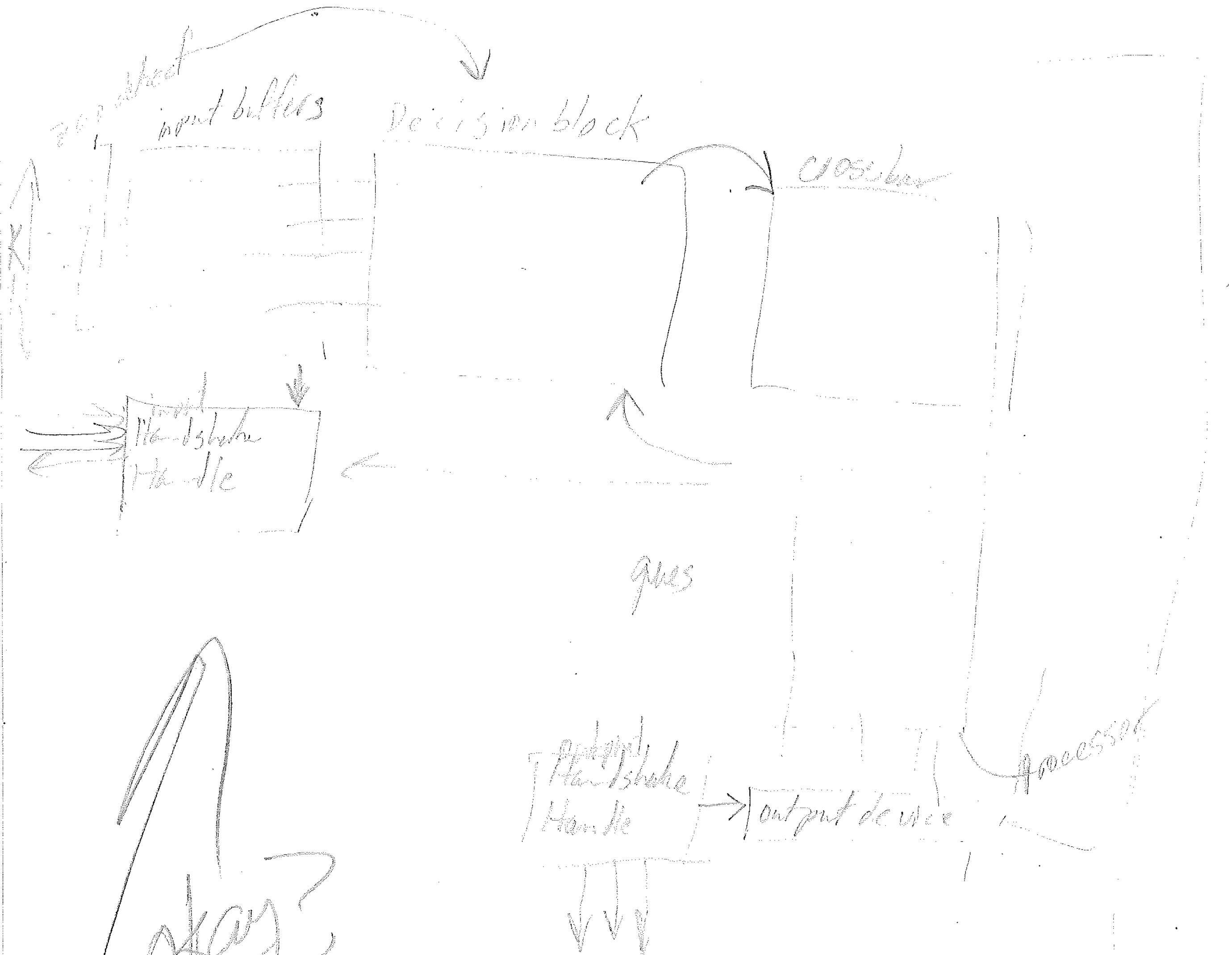
ground propagates up ~~until~~ a set is encountered triggering

use this one on high

Taft says is safe decision ~ 1 microsec

using this comparator for each que we need
extra bit for full or not
2 control wires { one with inverters → drive shifter / one without → drive comparator

OVERALL



okay?

Danny – left for weekend
(see you tonight or sun night or mon)

~~most~~ Handshake Handler asyncronous

see if can ~~take~~ message then arbitrate
must be conservative

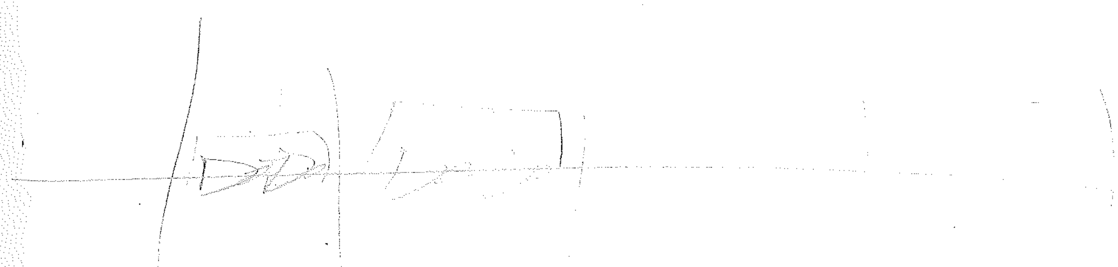
counts the not full (not filling) ques

counts input fulls
subtracts

if left over then decide between ~~options~~
options

~~most~~

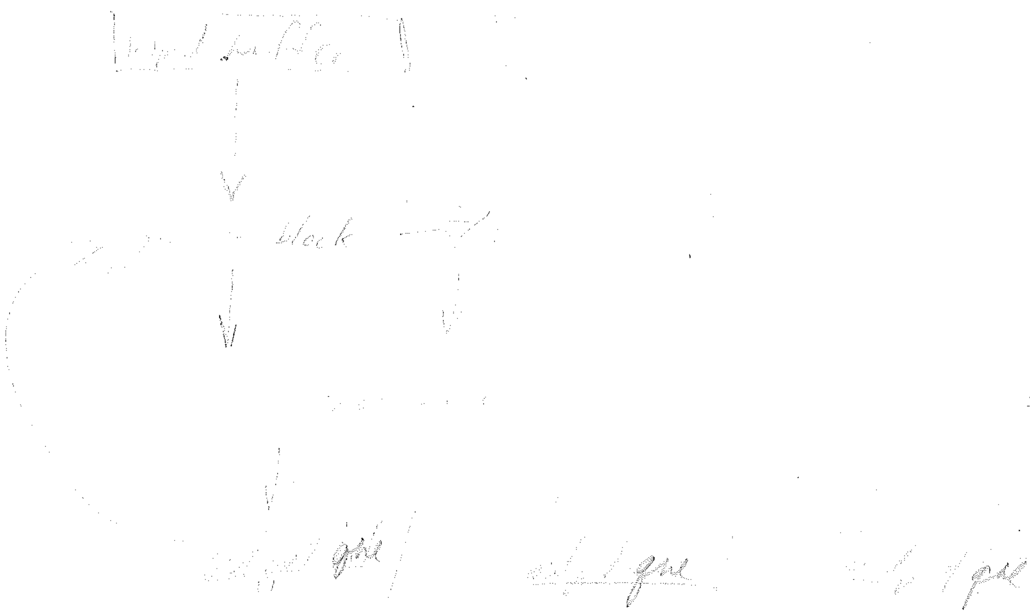
use a que and masks



input asyncronous
output syncronous

"ai:husli;que newcif"

Please leave this here

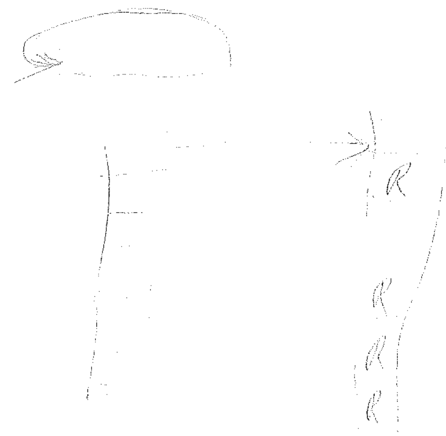

block

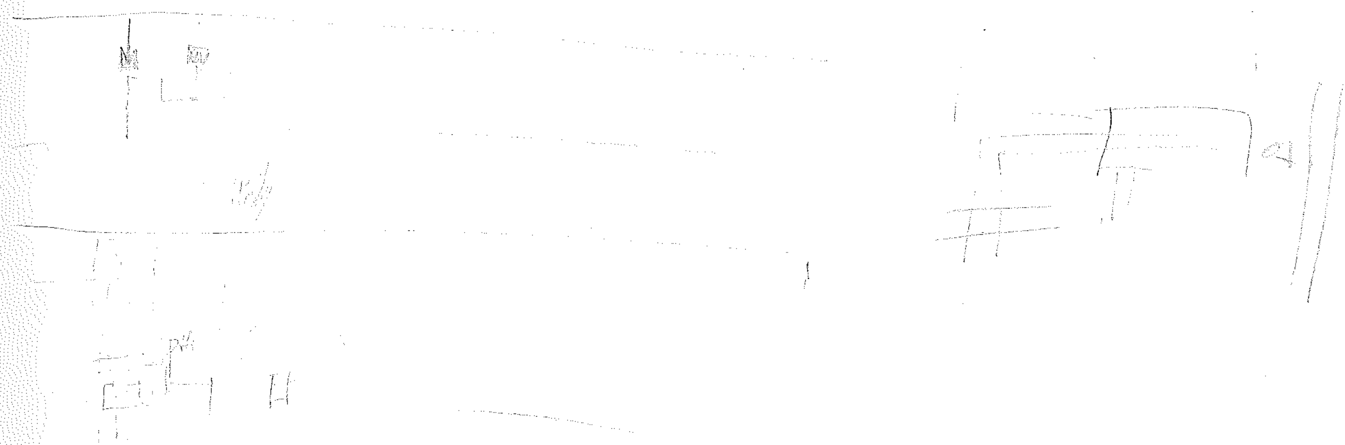
A 0 1 0 1 1 1 0 [illegible]
B 0 1 0 [illegible] 1 [illegible] 0

A⊕B C [illegible]

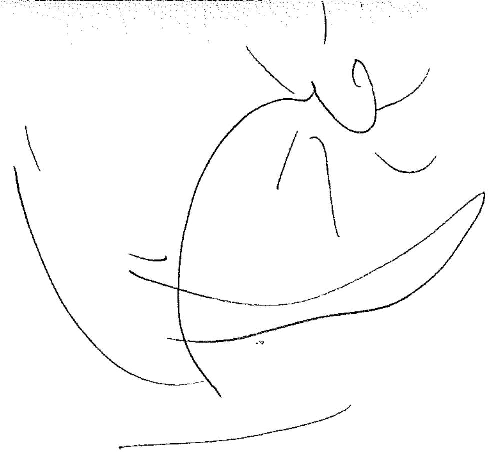
[illegible]

0 1 0 [illegible] 0 1 A
0 0 1 0 [illegible] 0 1 B
0 0 0 0 [illegible] A·B
0 0 1 0 0 0 0 1 C
0 0 1 0 0 0 0 1 C·B
0 0 0 [illegible]





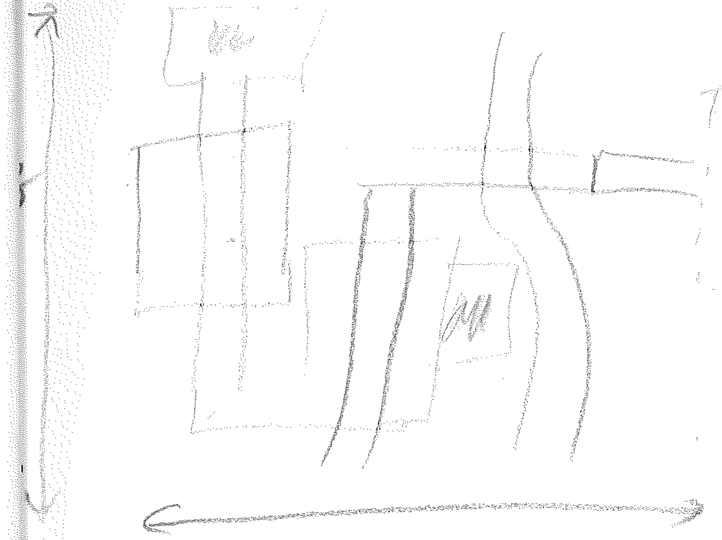
Length = (Length per ~~*Polysilicon~~
poly-diff tip to contact
*contact width wld



VTI R+W busses 7 λ each
2 bits control reg single dir shift

mine RorW bus
shift-regs double direction shift
1 bit control reg

shift on ϕ_1
write on ϕ_2



21

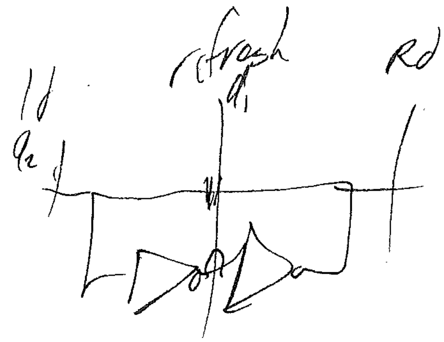
= ~~482~~ 483.

23
21
23
46
483

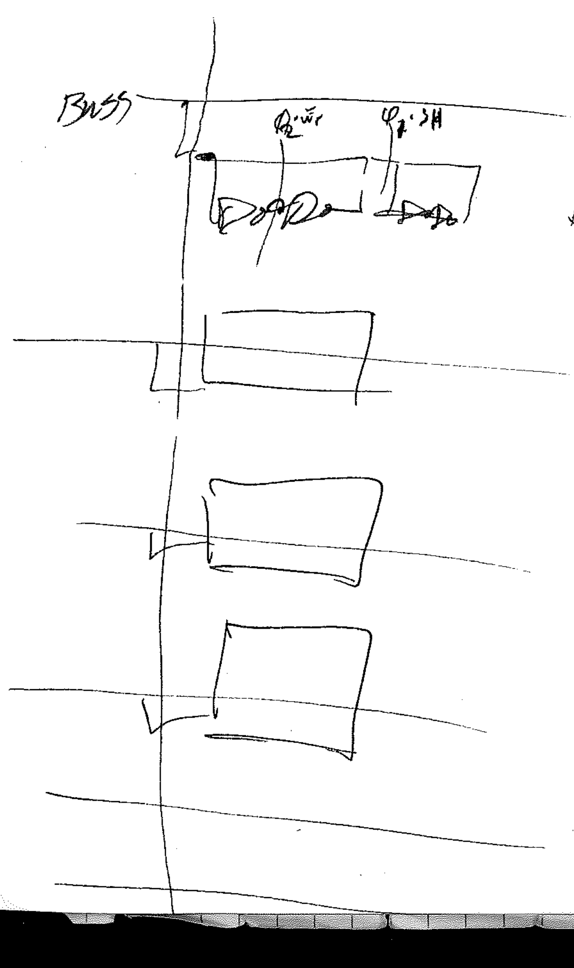
w + 1
L = 2

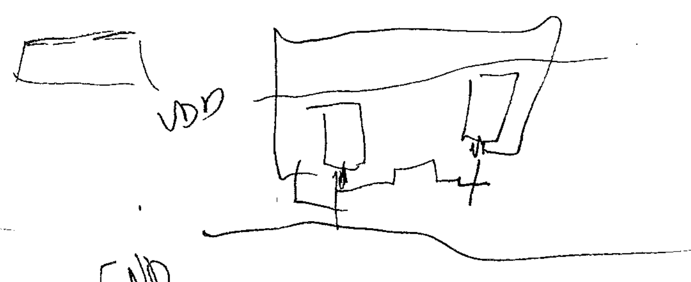
23
21
~~462~~ 483

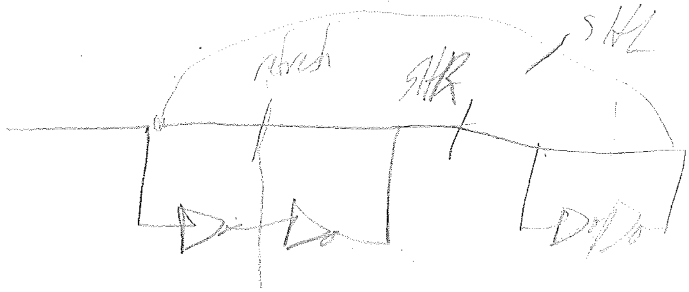
19
24
456



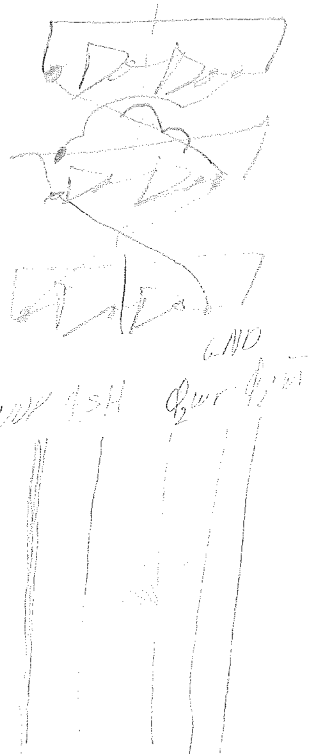
write here







SHR = SH·...

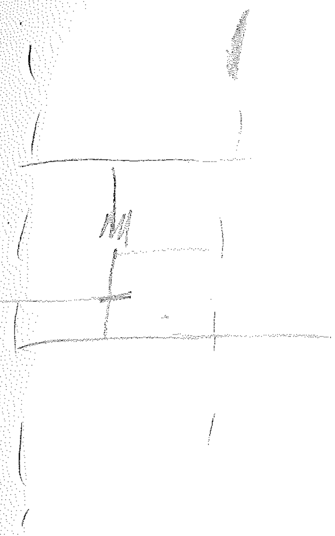


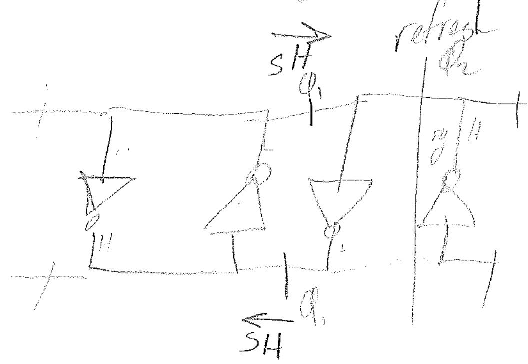
SHL = LD · SH

neither LD·SH + LD·SH

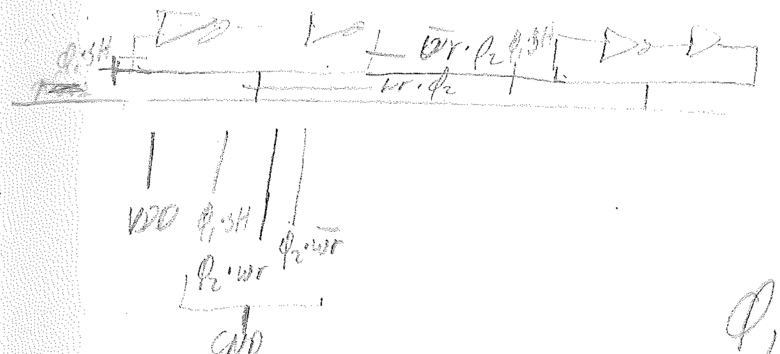
TAPE 2879

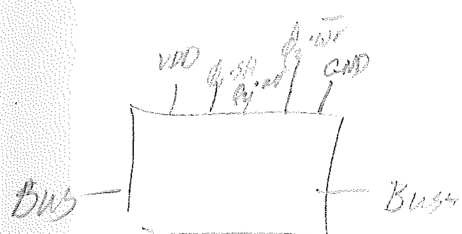
232 AR9 WORMS





Que cell





ϕ_1 = shifting time or not
ϕ_2 = writing time wr not

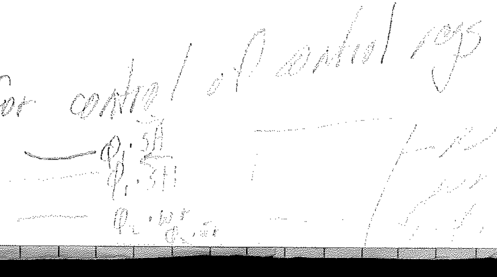
for control regs must be stable during ϕ_2

$\overrightarrow{SH} = \phi_1 \cdot SH \cdot \overline{wr}$

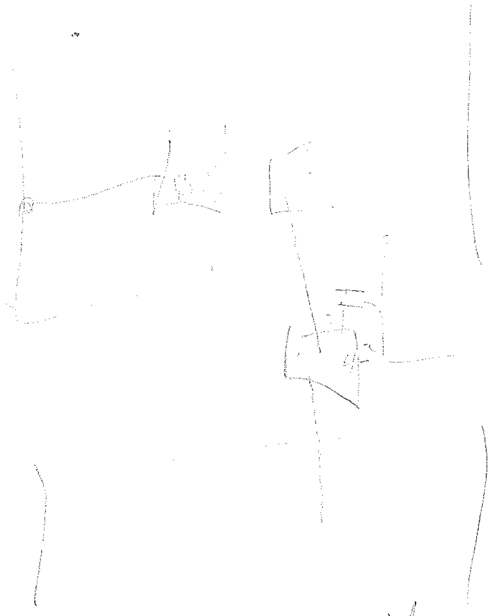
$\overleftarrow{SH} = \phi_1 \cdot$ had wr in last $\phi_2 \cdot \overline{SH}$

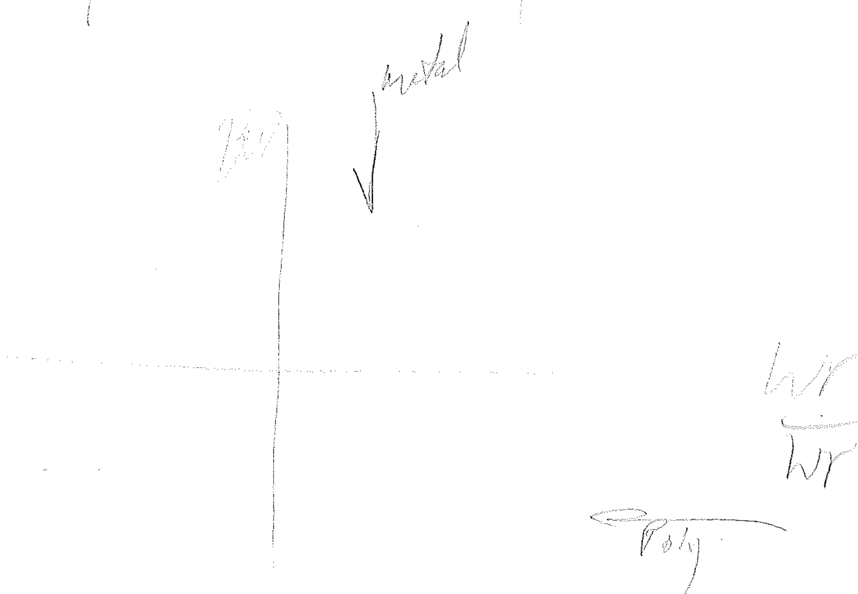
Neither $= SH \cdot wr$ or $\overline{SH} \cdot \overline{wr}$

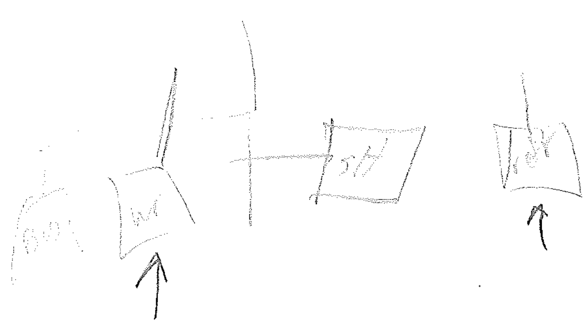
for control of control regs



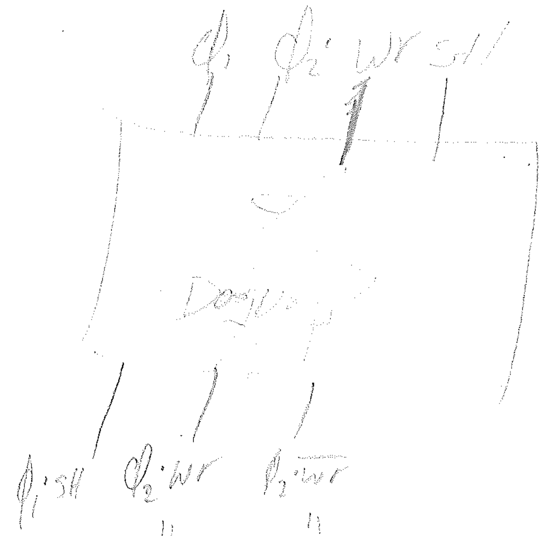
wr	reg	d.	wr reg d. wr	wr reg d. $\overline{wr}$
0	0	0		
0	0	1	0	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	0	0
1	1	1	1	0







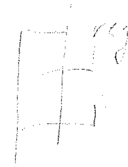
$\overline{A+B} = \bar{A}\cdot\bar{B}$

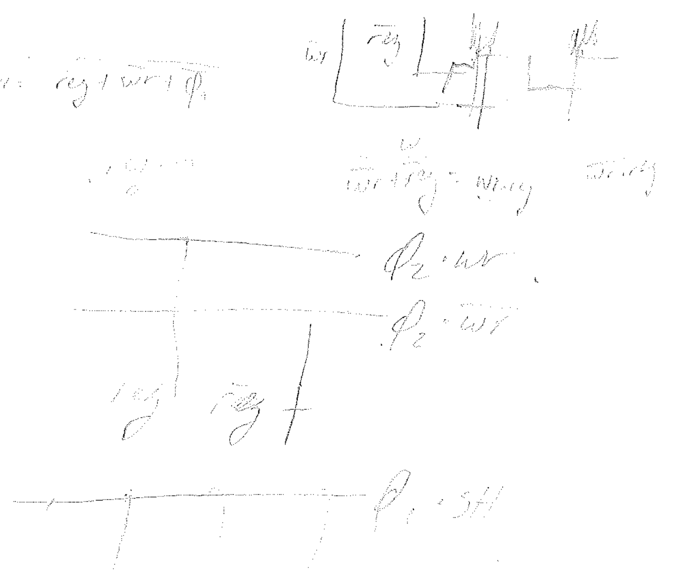


reg L ⇒ send H on ϕ_2
reg H ⇒ send $\overline{wr}$ on ϕ_1

reg · wr · ϕ_2 $\phi_2\cdot\overline{wr\cdot reg} = \phi_2\cdot(\overline{wr}+\overline{reg})$

—— wr · ϕ_2



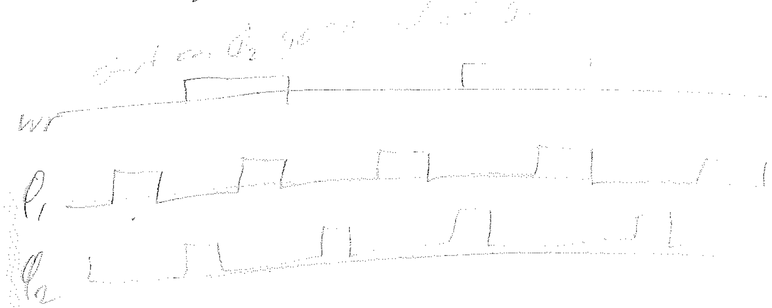


wr · reg · ϕ_2 = $\overline{\bar{\phi}_2 + \overline{wr\cdot reg}}$

$\overline{wr\cdot reg}\cdot\phi_2$ $\bar{\phi}_2 + wr\cdot reg$

reg · wr

1 or
2 or



A			
0	0	0	1
1	0	0	0
0	1	0	0
1	1	1	0

setnet

1F AR 9

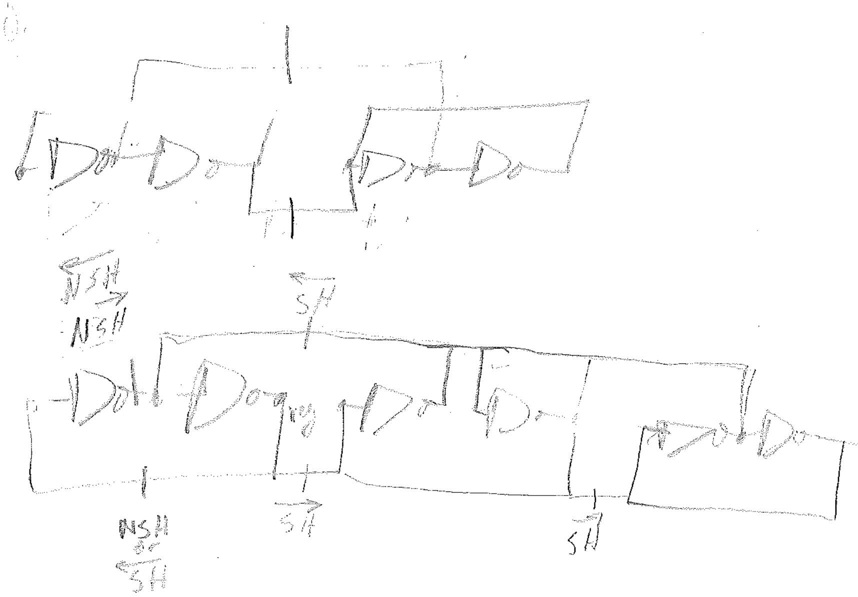
list of files

$\frac{L}{w} \Big/ \frac{L}{w}$ $\frac{6}{2}$ $\frac{6}{2}$ $\frac{2}{4}$

$wr \cdot reg \cdot \phi_2 = \frac{\overline{wr + reg} \cdot \phi_2}{\overline{wr + reg} \cdot \phi_2}$ $\frac{\phi_2}{wr \cdot reg \cdot \phi_2}$

$wr \cdot reg \cdot \phi_2$

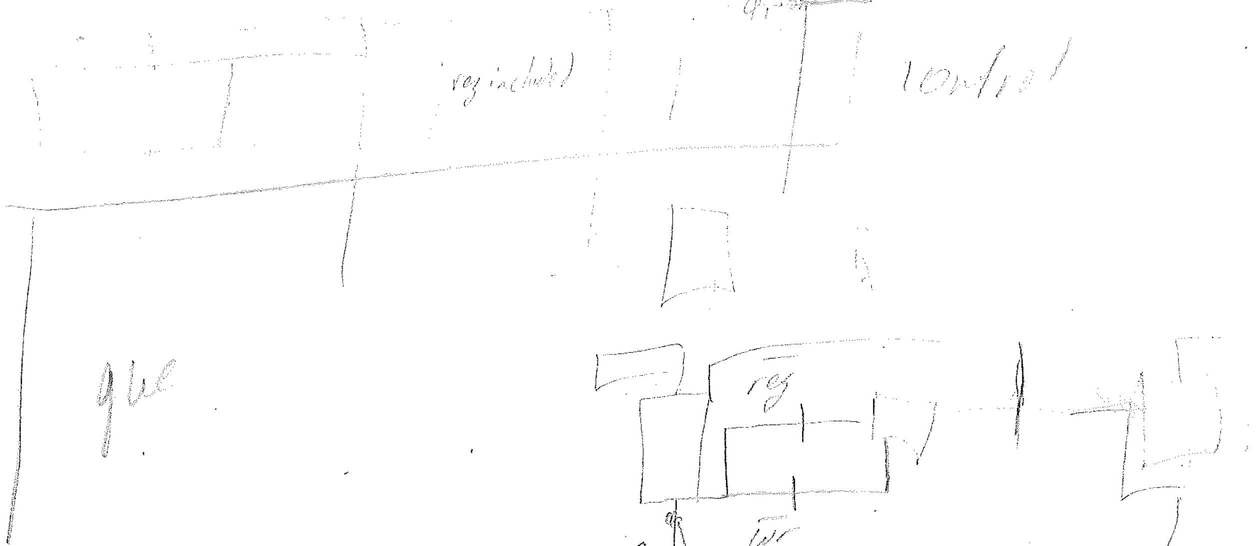
$\overline{wr + reg} \cdot \phi_2$



KA 14LE

~~Hall~~

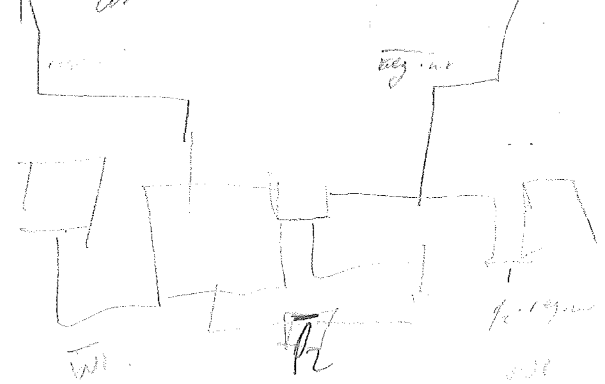
Holt 5129

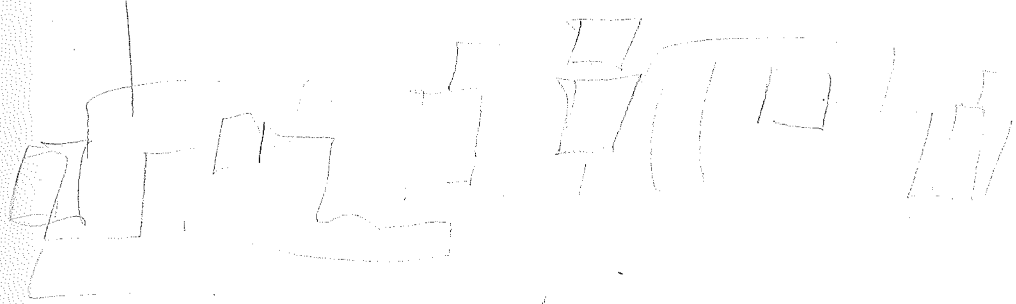


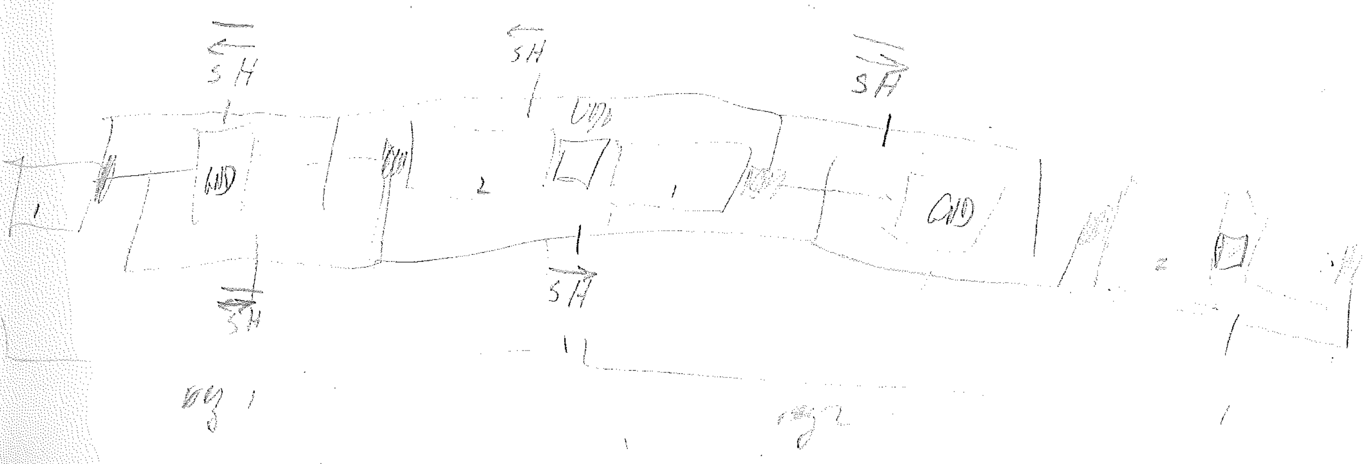
$Nor(Nor(\overline{reg}, \overline{wr}), \overline{\Phi_2}) = wr$

$Nor(\overline{Nor(\overline{reg}, \overline{wr}), \overline{\Phi_2}}) \cdot wr$

1 2 3







wba;dict letter ... intofile

Spell dictionary:

small
ml:wba;Dict > (with suffix attach... eg. ABBE/S/M
or [XX] <wbA> :Spell.Dict.
ABBE/S/M) not in order

Now in HUlsi: BKOUT

larger (subset of ?) ml:dict;mwpd List in order

large
ml:dict;mygel > (danny says)

to get word out of ^
Line starts with F and
ends with the word (after a numeric
def. ends with no...

pairs of num
c cycle-number ( )
arrival at end
1st num is how long it took
2nd is optional
q is worm put on q
Bug completely saturated
z typed when
cm: groved

(setq wordarray (make-array

(with-open-file (foo "――" ':read)
(loop for

Outright
Outstrip /s
Hallmark /M/s
Calvadry

spell
ver1

[illegible]
ver2

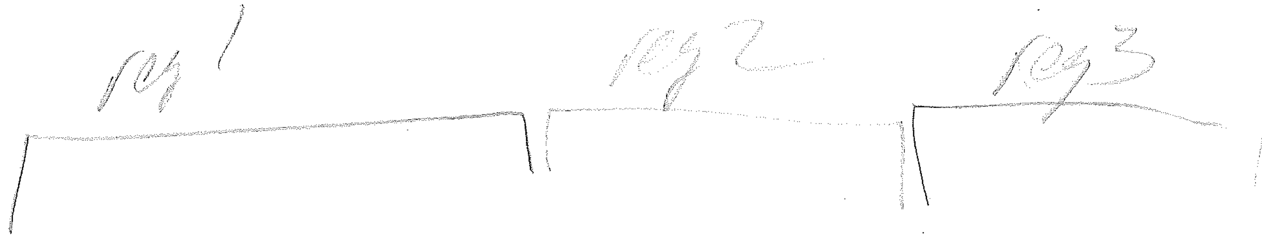
register control

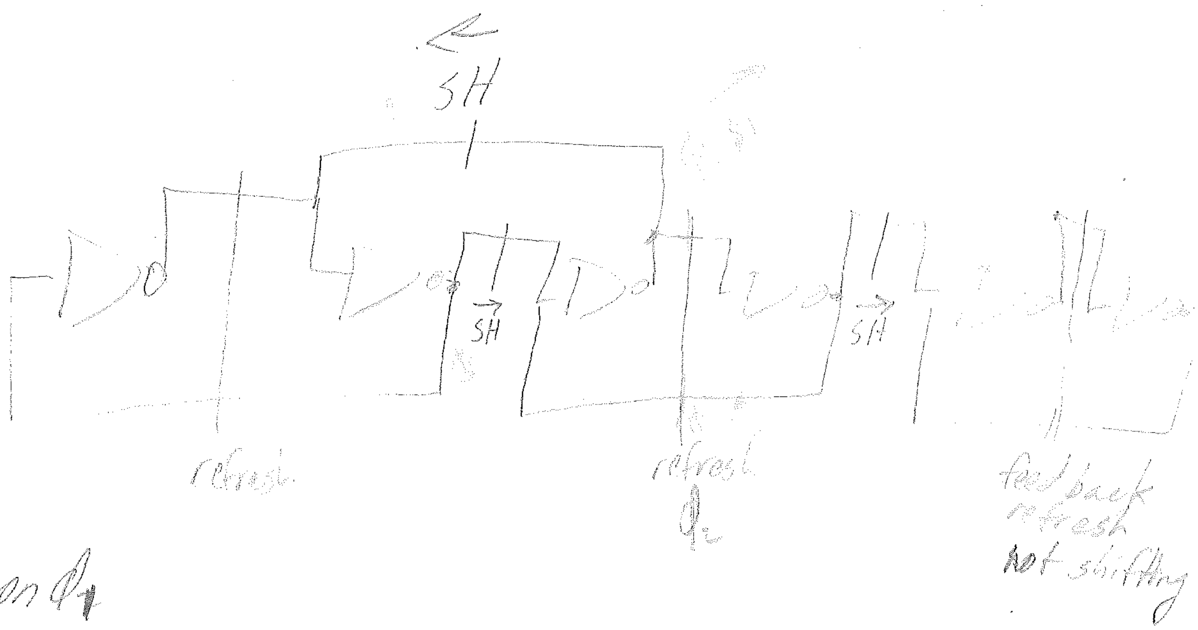
$\phi_{out} = \overline{\phi_1 + SH}$

$\phi_{shift} = \overline{\phi_1 SH + wr}$

$\phi_1 \cdot \phi_1$

$\overline{wr} \cdot \overline{wr}$



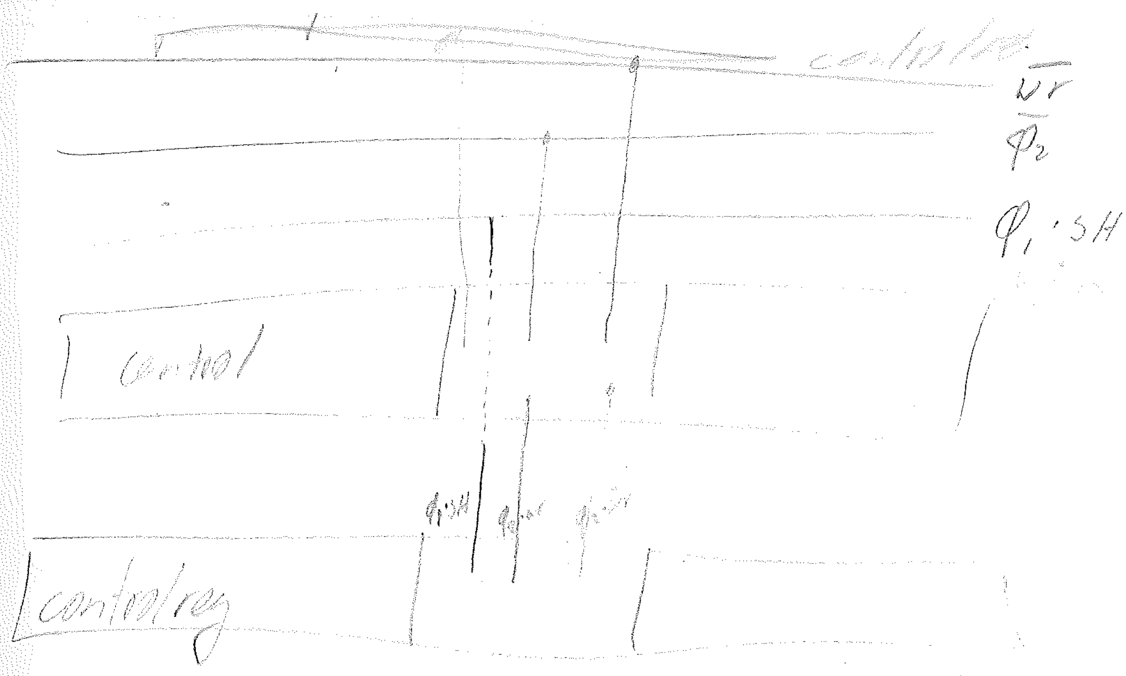


all on ϕ_1

$\overleftarrow{SH} = wr \cdot \overline{SH}$

write on last cycle ϕ_1 not reading out this cycle.

$\overrightarrow{SH} =$



φ_1

φ_2

wr

SH

$\varphi_1 \cdot SH$

"foo, bar baz"

Brewster Kahle

MIT NE43-352
545 Tech Sq
Cambridge MA 02139

4 lines .10 Fastline Bendix
din style

hypercon

specs:
17" x 12"
8 layers
6800 contacts - 17 rows of 400 contacts each 1/10" spacing
10 mils wire & spacing

qty. 10
Din connectors

connector people

Own connectors? yes reliability

send specs, sample

can you take others connectors

.10" ~10 boards 6" x 10"

Teradyne 603-889-5156 (x464 Fern Talbott) 8

make 4 layer inhouse

can order out more layers approx

they make co

x513 Harry Cook

Nasua NH

sent schematics
call him later or he will call
sending stuff

KS1005 Backplane

1/8"

Garry 201-846-5280 Bill Schwartz 16-18 weeks

Call back at 3:30 tues

IPS specs visit the

they will use other's connectors

not printed circuit finger

vary weight of copper

recommend a NAFI connect

.015 width conductor
.015 spacing } 2

sending quotation

For quotation; size of board # layers

holes

Tyco Backplanes John Seussel

Use Others connectors

sending rep

cannon
Winchester

See next page

EL FAB

214-233-3033 sales ~~Engin~~ dept.

~~will call back~~ will send info

Harry Weaver x249

have mass. reps:
CM Associates in Framingham Don Speil
617-872-4871 Dave Slote
1-

~~Datacom~~

Alco ~~890-~~ 969-2620 No go say they ~~Jon~~ don't do it suggested Angat

~~Angat~~

Elco 213-675-3311

Hank Horn x2347 I send
VG connectors Pranity
He's sending stuff

No Basic

Could do artwork

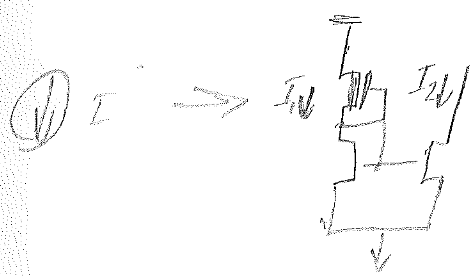
2250 Park Place
El Segundo CA 90245

Angat (bottom hand only) ~~Bill~~ Dave Maxwell will call back
762-3164

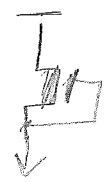
Punt

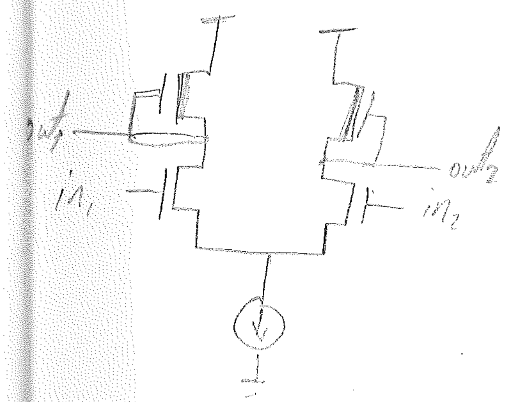
Comparator

Knight
for Bidirectional wire

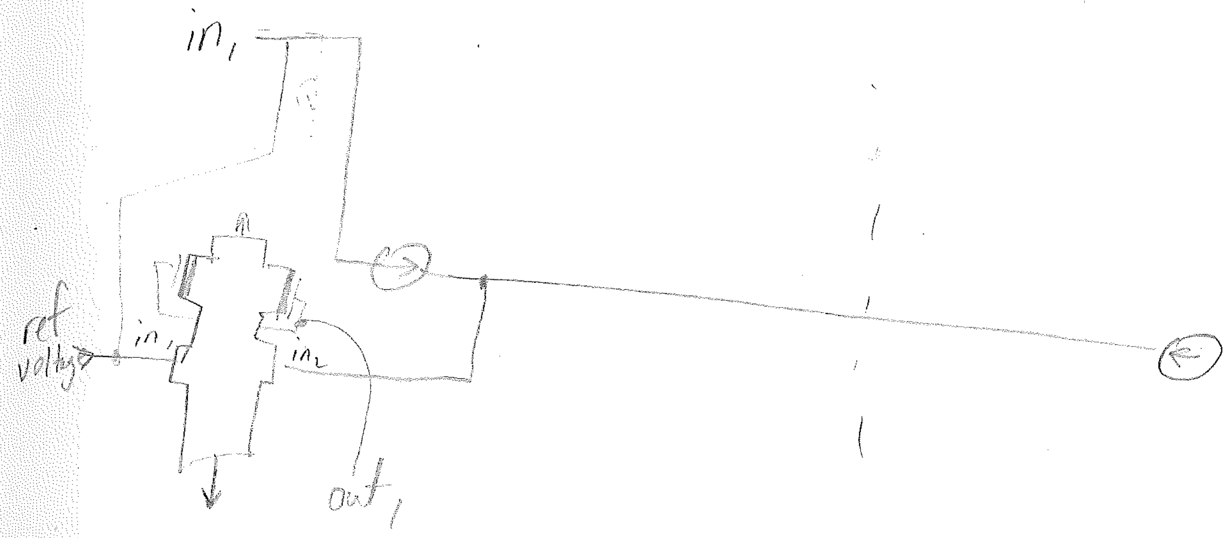


$I_1 \doteq I_2$ or





diff amp
good for comparator



Yeilds

assuming
chip 2147 is 80% yield

200 × 150 mils = .480 × .360 cm

$y = e^{A/k}$

$k = \frac{A}{\ln y} = 76 \times 10^6 \mu^2$

$\therefore y = e^{A/76 \times 10^6}$

$k \ln y = A$

if willing to have 25% yield then

$A = 1 \text{ cm}^2 = 10^8 \mu^2$

proc/chip:

140 bits storage/cell

1/2 shift reg = $23 \times 17\lambda = 391\lambda^2$

= $109K\lambda^2$ /cell

×64 cell/proc = $7 \times 10^6 \lambda^2$

= $28 \times 10^6 \mu^2$

For gue cell

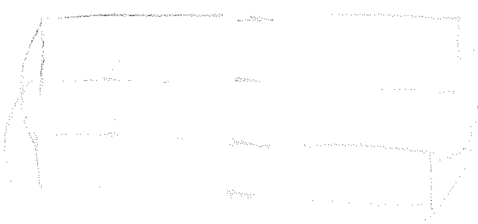
$\sim 1000\lambda^2$

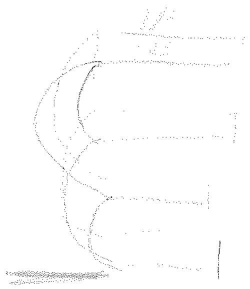
64 RAM / chip
8 x 8 ~~chip~~ /board 64
256 boards

128
64
512
768
8192
64
1768
112
72296

4X2 · 2 ... = 256

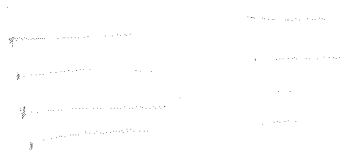
1A

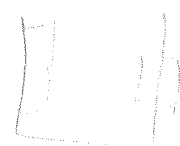




4X2 X

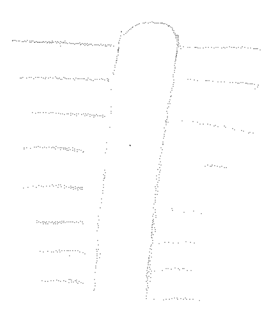
3 chips





$1m = 2^{20}$ cells
2^{14} chips
2^{7} boards = 128
[illegible] chip
1/8 of 1
machine
16 [illegible]

16



Wi.ch.ta
(Joe Levine Con

Tyco
10 layers [illegible] production

[illegible]

10 mils wire & spacing
[illegible] connectors
pressfit [illegible] & solder

inhouse 30" 30"

Donot do layouts

[illegible], density, # of connectors
line width & spacing
[illegible]

# of layers, copper weight

2 oz power 1 oz

10-12 weeks

~~[illegible]~~ 6 weeks
premium

Bare board test
Final test

$1.50 Test fixture

Tyco meeting 7/21/81
Mark Newby
617-746-3869

{ ≥10 layers are in current production
{ 16 experimenting with them

{ 10 mils ~~spire~~ width + spacing standard
{ 8-6 is pushing, still possible but high fault rate ∴ price high
(rather see more layers)

connectors: they are most confident with winchester
will use anyones.
He ~~[illegible]~~ was not familiar with Din connectors, will send
winchester lit. on it.
recomends pressfit rather than solder

wire wraps option: unsure if possible with din connectors

maximum board dim ~~30~~ 38" (-½") x 30" (-½")
leeway

They do not do layouts. There are firms that do.

Takes 8-10-12 weeks for a run. 6 weeks for premium cost

Testing: 2 types 1) bare board test, they make a test bed (of pins)
to test each interconnection on a board
cost: $1.50 per hole usually not done on prototypes

2) Final test test for shorts + open with connectors on,
will tell us of problems, but they ours

No problem with current tech to do 17"x12", ~~10~~ layers ~~$~~ 16, boards, ~~[illegible]~~
2 wires ~~between~~ between, 800 contacts/board
Will: phone estimate, send winchester + other (if any) over

impressions:

solid person, ignorant of din, willing to help work on half cooked specs.

To get quotation:

1) dimensions of board
2) wire spacing + width
3) # of layers
4) # + Type of connectors
5) weight of copper 2oz for power + gnd } common
   1oz for signal
   10oz symbolics (for some reason)

6) extra parts on board
   power + gnd lugs

AMP connectors

1000 series Box connectors

lower insertion force

879-6850 Mit rep

product tech Brian Mahoney 215-647-1000 x469

List machine boards

36 lbs insertion force

500 series

can be put into multilayer boards

How to program Comm Chip
easy: have a word who's bits map to the options
load during Boot

hard: self optimizing (Bla)

have gues signal processor or something .. they
were used

easiest implementation:
have all options there. some can be
turned on or off, just power or not

harder: Floating gate cells that could be assigned to
input, output, processor

easy 2 deep gates at input & out and ~~[illegible]~~ proc
3 deep for processor

Comm Chip

options

ques: no que
input que
output que
how deep
floating que
how many (programable?)

internal word length (this better be fixed)
constraint: delay on chip is fat least)
cycles = word length / buswidth
= 20 / 2 = 10

DecisionBlock ~~[illegible]~~ options

determined direction based on address ∴ output or input que
eg go to correct x then y

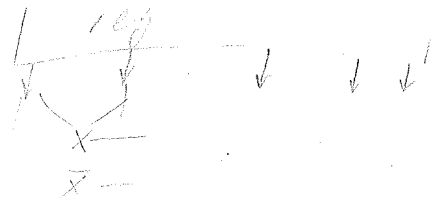
choice of directions based on address + state
eg between all advantageous directions
output que optional
∴ input que optional

B

TO decide if going for 0000 or 1111
must take top k+1 bits and and determine how
zeros or ones

Digital comparator
increments, decrements

given k=2, ... comparison
4 or 6 bits



Daughter Board
quantity 16
4 or 6 layers

1200 pins on connectors

~~8x~~12"x20"

68 pins/chips X 8X16 chips

Automated Artwork Generator

Airborn Dean Remillard 214-931-3226
Connectors

<u>only 3 row</u>

sending stuff

The pin is max
engage force 5 lb avg 2.5 lb
with max .01 x # contacts force

withdrawal .019 steel pin → .45 oz

engage .0315 steel engage .12 oz

?
it is .12 or lb

Artwork people

Computervision 215-1800
No Go

Applicon wrong route

Electromask No Go 213-884-5050
Paul King x144

Photo sciences
Dept 6 19223 S Wilson

Macro Graphics
Less Woods 213-701-1004

sent 8/4
Send him info to see what
20416 Pirie St
Chatsworth CA 91311

Graphic Engineering
Then
Take "Heuberbergs" tapes puts out master Artwork

Photo Circuits
Dave St Andre will call back
516-488-1000
about PC boards

Automated Images Inc
Mike Wilson
components on daughter
pins on mother

Dont put i.c. or on bus on design

Will send quote

take pin data and do cost on both daughter & mother

multi wire — wire to wire for Bolzano
16 mil [illegible]
insulated
6 mil ~~[illegible]~~ copper magnet wire
sending stuff

10 layers => 2 or 3 layers

Can make 5 layers

300 soldered 5[illegible]

Paul ~~[illegible]~~ Lee 603-889-0083 ← call later
regional sales manager
Nashua NH

41 Simon St
Nashua NH 03060 send footprint of chip

6 weeks [illegible]

110 [illegible] [illegible]

[illegible]

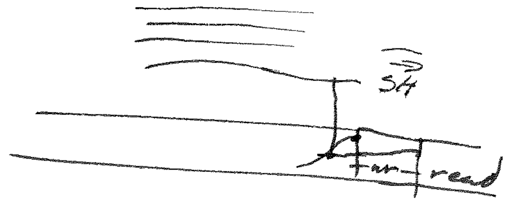
Amp (connectors) Inc
1-717-564-0100
1-215-647-1000 x499 Government office
Tom [illegible] will call us

(-434, 0) (39[illegible]

Winchester tyco connectors
meeting wed 1:30
@ Joel Levine ~~203-274-8891~~
617-256-8791

3

outputs ..., write, read, ...



inputs
= read
= write

$\varphi_1 \cdot \overleftarrow{SH}$
$\varphi_2 \cdot \overleftarrow{SH}$
$\overline{init}$
$\varphi_1 \cdot \overrightarrow{SH}$
$\varphi_2 \cdot \overline{\overrightarrow{SH}}$
$\overline{wr}$
$\overline{\varphi_2}$
$\varphi_1 \cdot \overline{\overrightarrow{SH}}$

$\varphi_2 \cdot wr \quad \varphi_2 \cdot \overline{wr}$
$\varphi_1 \cdot SH$

$\bar{r}\cdot\bar{w} + r\cdot w + w\cdot\bar{r}$
$\overline{\bar{r}\cdot\bar{w}}$

$\varphi_2 \cdot \overline{\overline{SH}} \cdot \overline{init}$

$\overline{\varphi_2}$, $\varphi_1 \cdot \overrightarrow{SH}$, $\overline{wr}$, reg

$\overleftarrow{SH} \cdot read \cdot \overline{wr} = r \cdot \bar{w}$
$\overrightarrow{SH} \cdot wr \cdot \overline{read} = \bar{r} \cdot w$
$NSH = read \cdot wr + \overline{read} \cdot \overline{wr}$
$= xor(read, ...)$
$\overline{SH} = (\overleftarrow{SH} + NSH)$

$\varphi_2(\overleftarrow{SH} + NSH) = \varphi_2 \cdot \overline{\overrightarrow{SH}}$
$\varphi_2(\overrightarrow{SH} + NSH) = \varphi_2 \cdot \overline{\overleftarrow{SH}}$

start carry = 1 inc

Data	internal carry (inc)	Data	carry
0	0		0
1	0	1	0
0	1	1	0
1	1	0	1

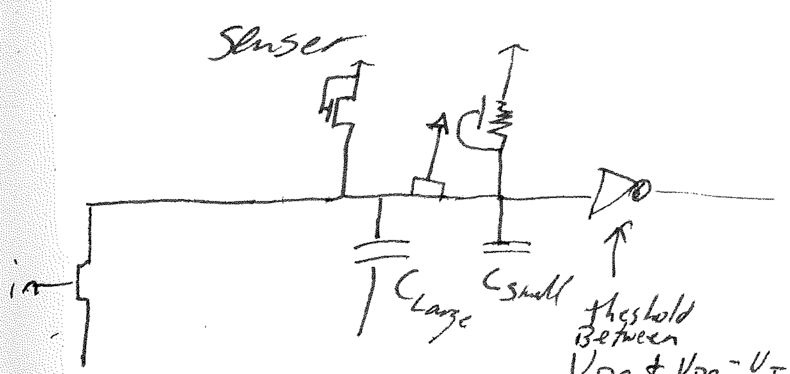
carry start 1

Data	carry (dec)	Data	carry
0	0	0	0
1	0	1	0
0	1	1	1
1	1	0	0

$N\ Data = OData \oplus carry$

inc $N.\ carry = ODATA \cdot carry$

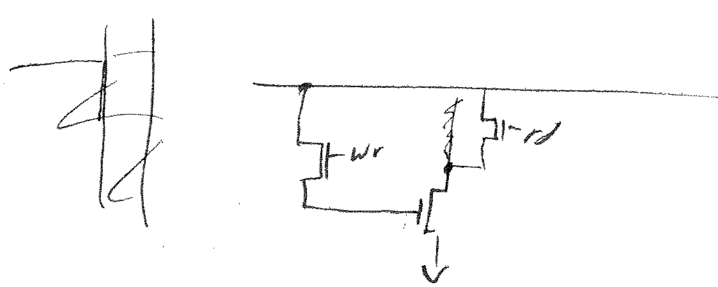
dec $N\ carry = \overline{ODATA} \cdot carry$



$$V_T = V_{T0} + \gamma\sqrt{V_{SB}}$$

Body-source diff

3 transistor ram



DO

glue (6 hrs/simulation) ~~KC~~ Batali

RAM (straz pulling it)

Incrementer design

Dictionary

Multimullic

mother tooling artwork 11,650
etc.

table load ~~[illegible]~~ 1600.

2 pieces 1300 each

3 pieces 1000 each

10 p 560 each

No assembly work

tell if power to gnd shorts

23.5 length board

2/10 each end

Daughter tooling 16,450

table load 2,000

15 pieces 727 each

150 piece 628 each

Aug at

Burium copper + Brass

wirewrap 3 levels/pin

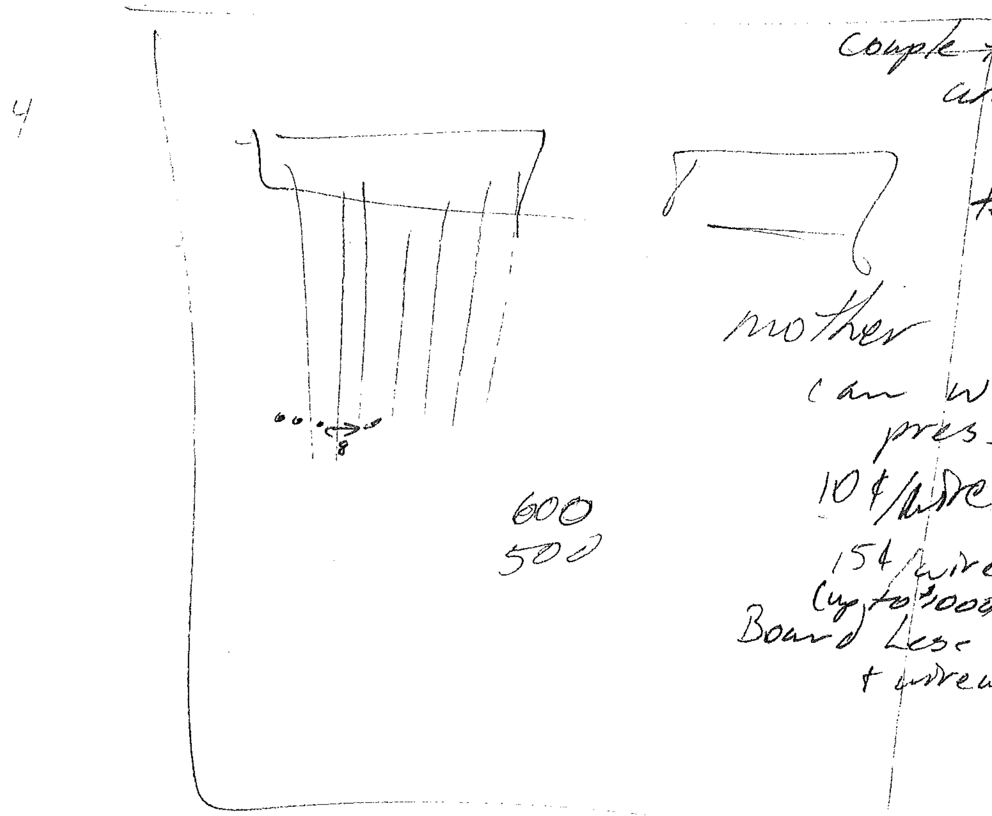
they do drawing

2 weeks design

~~???~~ 2 weeks

daughter board probably good
22.5 x

couple ~~thousand each~~
around $400 $200.



tes

mother

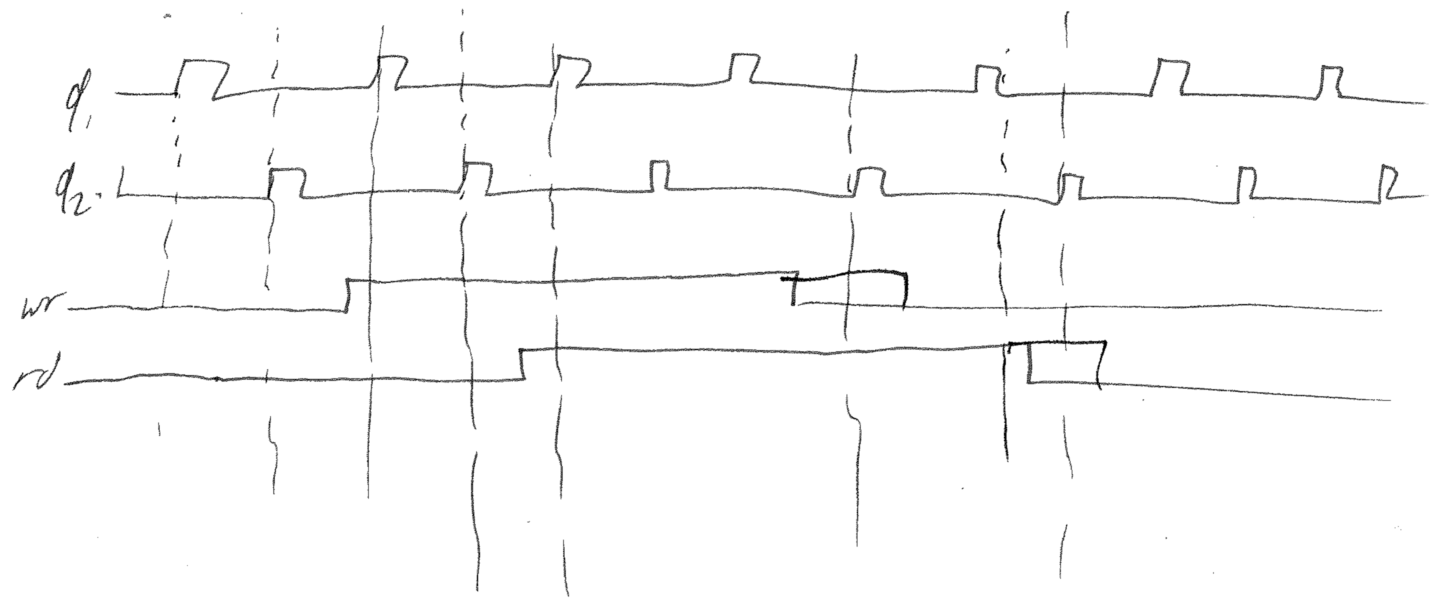
can wire wrap to
press fit

10¢/wire firm

15¢/wire programming (1 time)
(up to $1000) Design (1 time)
Board less connectors
+ wirewrap $500

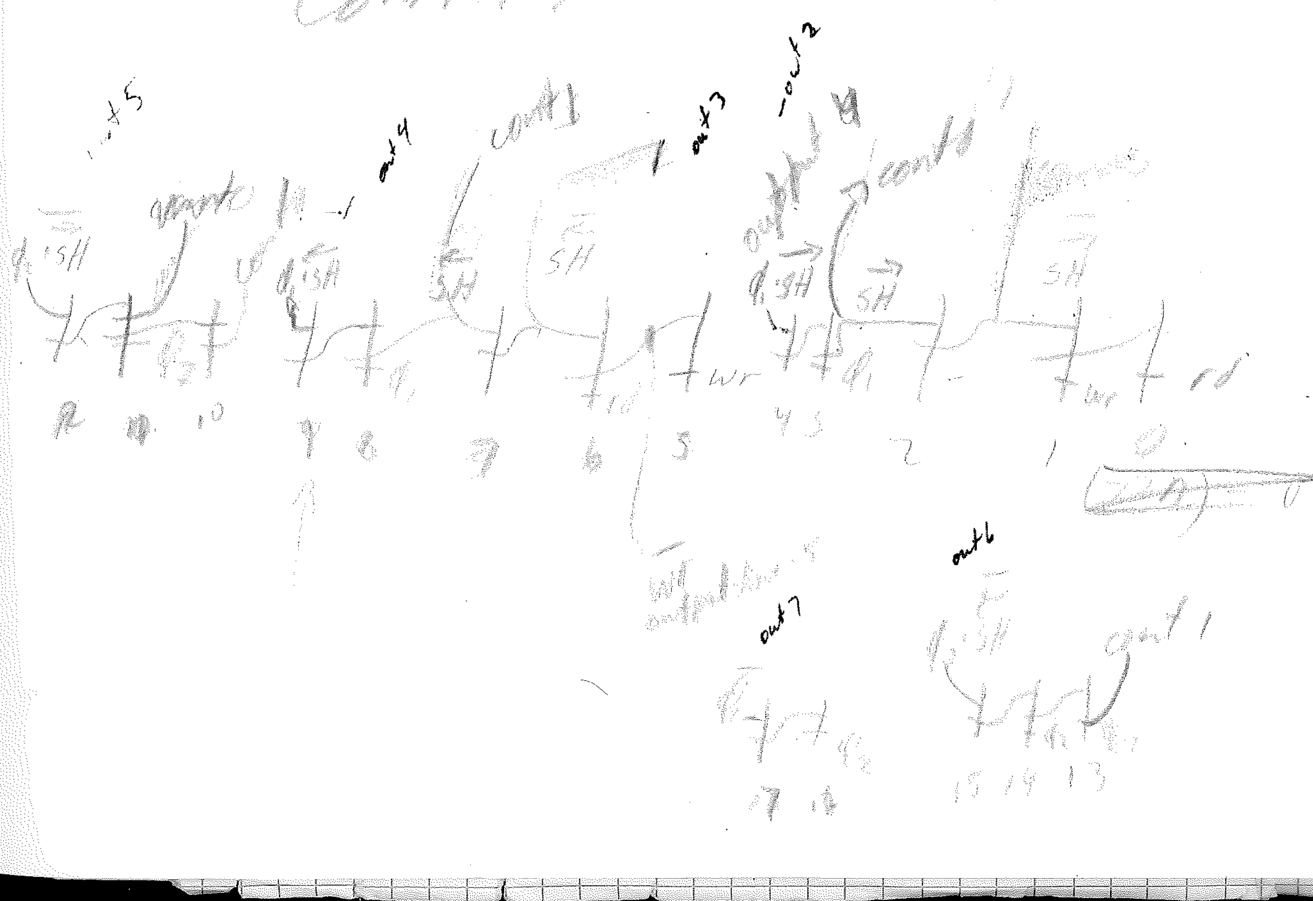
testing for opens

que[illegible] protocol



rise before ϕ_1 (after last ϕ_2)
fall after ϕ_2 (before next ϕ_2)
(believe

control inverter array



speech
dpl.nlm_t

cd lsi

(break V)
dir   directory

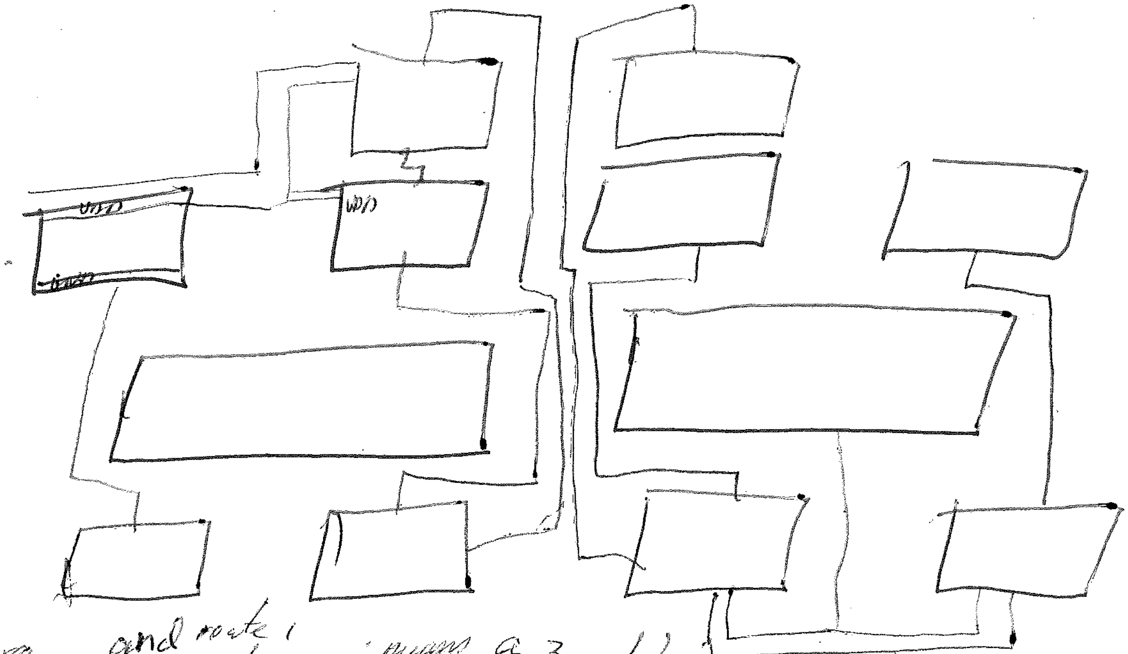
cd <lsi.lsiaa>

type lsiaa.lpt

LSIAA.Press

ftp
connect ai
ascii
get forsyn.file

disconnect



structure and route 1
route 2 ; means a 3 ended wire

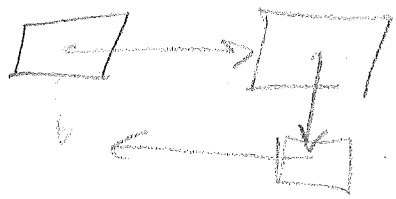
module interaction { hug-around-corner
around-corner
up-to assumes closest part in that direction
right-
down-
straight-connect assumes river or permutation connection between obvious points

wire interaction { top level constraints:
(Do-not-cross <cable-name>), (Do-not-cross :anything)
(can-cross <cable-name>) can-cross :anything

?? around corner ~~[illegible]~~ <cornerspec> <cable-name>
yuck

VDD+GND always in metal, with width

Mod Placement

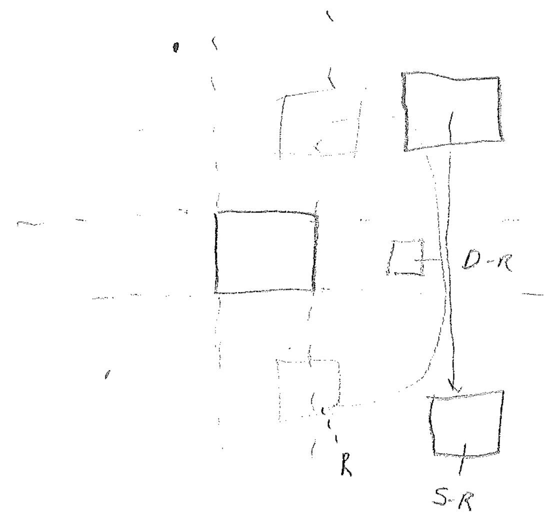


bl 1 S-R bl 2

bl

(SR bl1 bl2)

(DR bl1 bl2)



S	D	U	
A	⌐	D	
		R	
		L	
2	2	4	16

defconstraint 'optional-name
align-top block1
(?? input1 block2)

def constraint 'optional-name
block1 to-right-of block2

defblock (PP ((pp-object Bla (contact1 b-node1))
constraint 'const1 next-to-right-of block2
party inputs cable
contact1
contact2
cable3
party output wire
contact1

defwire (PP (number-of-them
from (?? output block1)
right-to block4)
right
:
TO (?? input block2)

Fonts on color screen

(funcall color-view ':set-font-map (list tr8))

smallest

(or Bigfnt) bigger

(format color-view "foobar")
prints foobar on color screen.

window info
Lmwind; window ~~window~~

DBL Draw

win for reviving → (Internal-based startup)

Files

~~BLK Com~~

BLK Bul

BLK Bas

BLK INIT

BLK Wind

BLK Com

PRIMOB prototype

Database

Jobs

→ Buffer structure

Daed stealing

DPL:Default-layer-list

flavor Bucky-Pane

in Daed startup: screen-Rounding-mode

Bucky-table

Daed-color-screen

in Do-it

In Internal-daed-startup the-empty-list?

;method Daed Pane :combined init

is in InitDaed

Daed-interaction-pane

Window-Command List

in BLKBAS

BLKCOM

Daedalus-running

&rebuild-types

buffer-list

View-list

color-to-LW

Screen-list

Setf ?

refresh

in method daed-status-pane show-status

View-name

cell-value?

assigned?

:property poly-connect

Draw-box

Diff-or-find

DPL:Def daed com

BBox-from-instance

View

Button-view-list

pack up parameters in LLKCOM

put "Block" in the
Layer-defaultsize,
& Default-size-for-layer-2
← Run-Rule

Interpreter in "OPL;make?"
make-a make-named

```
(Progn 'compile
  (Defun (Bla Maker-function) (*me*)
     (Party ...))
  (Deftype-Block 'Bla
          '((Pre-proc beo-pre-proc)
            (Post-proc beo-post-proc)
            (Primary-Params ((Leng 7)(width 7))))
          (Get 'Bla 'Maker-function))

  'Bla)
```

cont-part 1

left Rt

out $\overline{nreg + nwr} = reg \cdot wr$ $\overline{reg \cdot wr}$

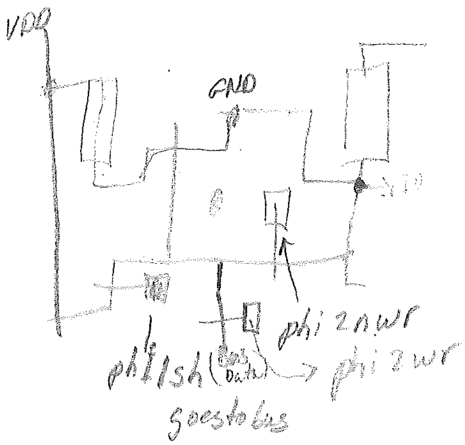
cont-part 2

out $\overline{reg \cdot wr + \overline{\phi_2}}$ $\overline{reg \cdot wr + \phi_2}$

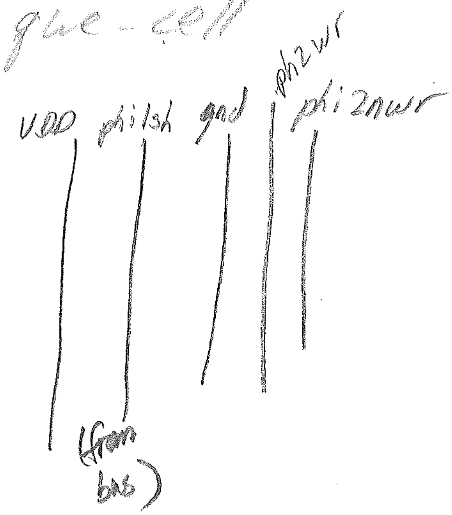
$= \overline{reg \cdot wr} \cdot \phi_2$ $reg \cdot wr \cdot \overline{\phi_2}$

que documentation

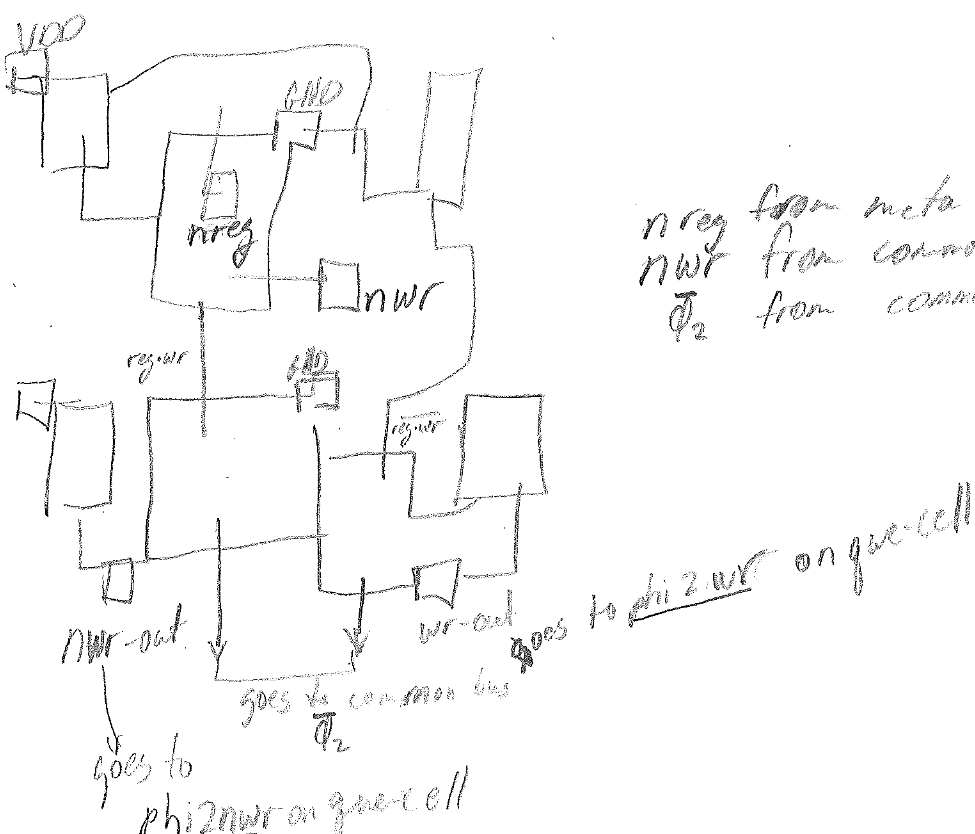
que-cell-without-metal



que-cell



cont-part 2



nreg from meta control
nwr from common bus
$\bar{\Phi}_2$ from common bus

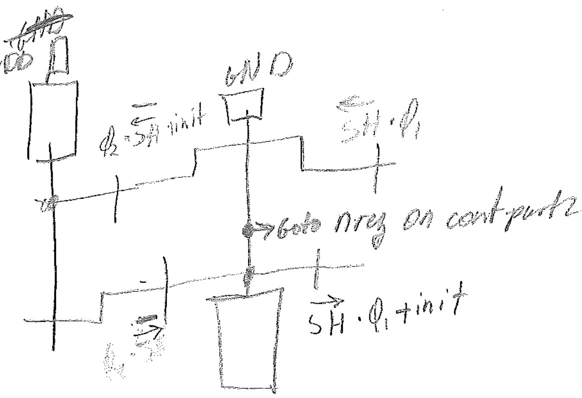
$\overline{rd} \cdot \overline{wr} = \overrightarrow{sH}$

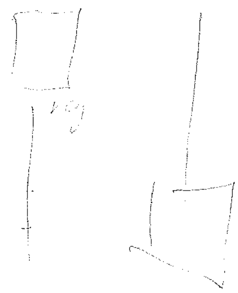
$\overline{\overline{\overrightarrow{sH} \cdot \phi_1} \cdot \overline{init}} = \overrightarrow{sH} \cdot \phi_1 + init$

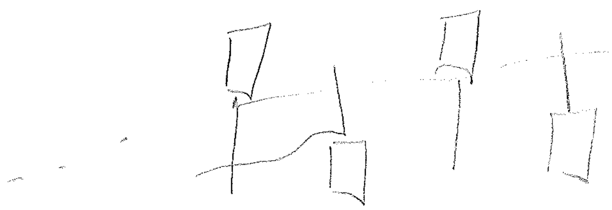
$\overline{wr} \cdot rd$

control-reg-cell

shiftonphil





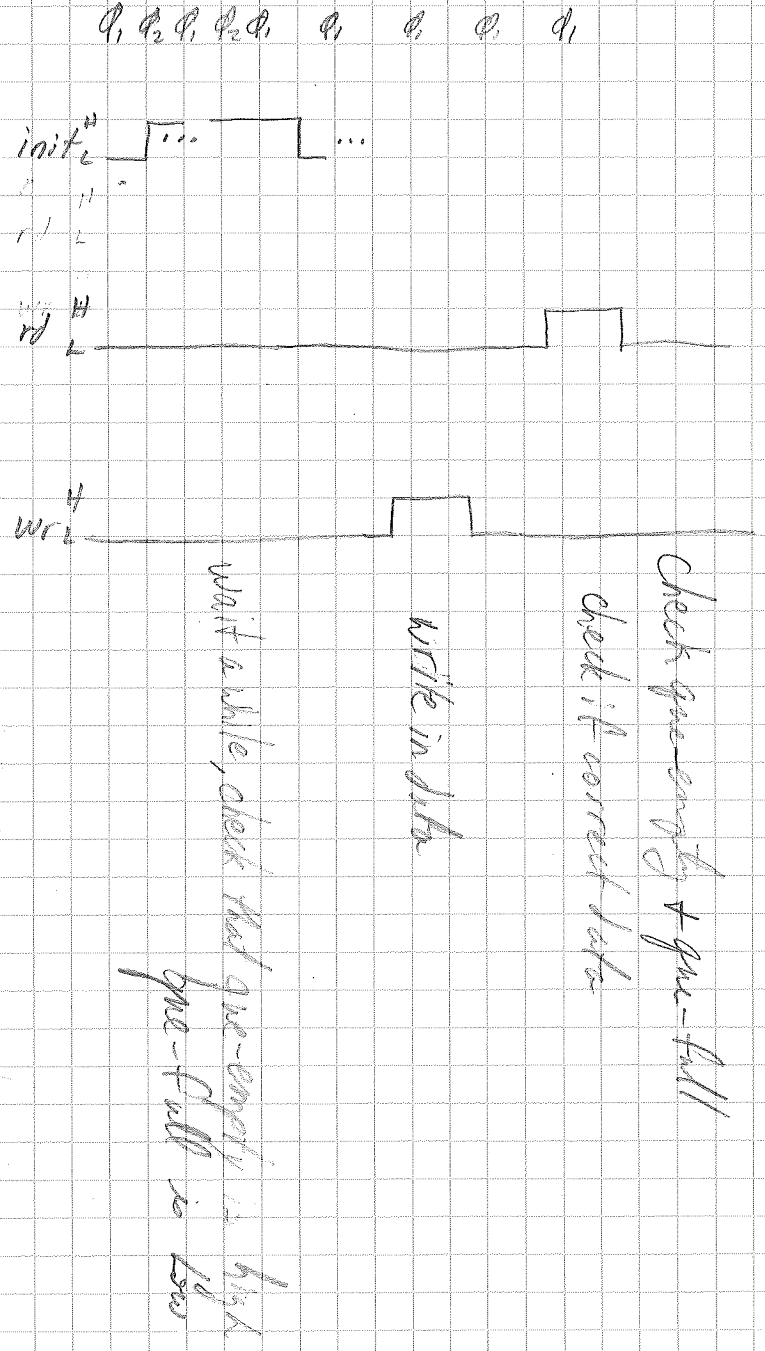


This array initializes by filling the registers with Highs.

Then it fills up by shifting ($\overleftarrow{SH}$) in Lows.

$$\overrightarrow{SH} + NSH = r\cdot\overline{w} + \overline{r}\cdot\overline{w} + r\cdot w = \overline{\overline{r}\cdot w} = \overleftarrow{SH}$$

To Test Chip



repeat for longer ~~duration~~ write
to ~~test completely.~~
go to que-full

To Test chip

ϕ_1

PI

(load "router; pi") or pi(k)

~~xxx~~ in (router; piman doc) is an short example

in deed
under extra-fet
(partg a a two-fets)
(prepare: router (>> aa))
(run-router a a)
(convert problem (>> aa)
refresh menu

~~xxx~~pi ~~s~~ample > is good test

(gc-on)

~~tes~~

(?? connection-points main-regs)

(the role-names (?? next-state))

schip3; hod ?

~~(load "IC; unpath init")~~

(load "schip3; dpath defsys")

test-main-regs)

indexes

pindary

mi

For TI to make it:

accept advise

asthetics

incremental

1. add modules slowly
2. have monotonic behavior

ie when moving two blocks closer
have them get harder to route

3. Have it ask to reroute if changes in big plan
are changing due to tweaking.

Tell where extra space is (for hand optimizing)

Hg
(hand-wire or not) data-enable-lines

clock-circuits
bottom-pads
data-bus-routing
pin left-external-bus

Fab obj

RPREP

(initialize-plas ())
if "field" error

Do (hardwire gnd-wiring)

(print shame 80

fake objects

Butali; Fake >

~~Fak~~

(load

(*path (tmp obj))

Fakeobj defcells

parts of scheme

load in order

butali; lmutil qfasl

(load b; "zwei"

(load b; "sys"

butali "plas >"

"schip3; sysdef >"

<super> E ~~compiles~~ evals buffer

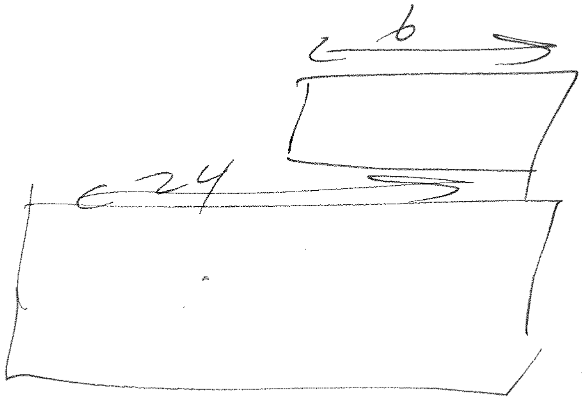
(:load-system fake

?(:load-system plas)

schip3; hog placement & routing of vbacnd

inhog (schip-part name (real-part fake-part)) depends on *faking-it*

layout is def of 2nd layer down

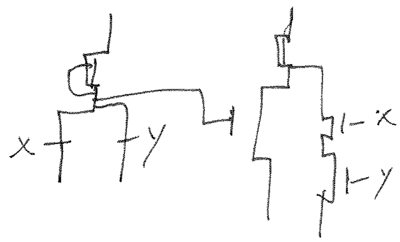


diff-metal 14Ω/μ²

metal cap 2×10⁻⁵ pF/μ² or 5×10⁻⁵

diff cap area 7×10⁻⁵ pF/μ²
periferal 20×10⁻⁵ pF/μ

Xor



6μ × 1000μ
6000 μ²

metal capacitance

(for 3 trans run

6000 μ² metal

24 μ periferal dif × 24

36 μ² dif × 24

8 μ

C
12000 ×10⁻⁵ pF
4800
10000

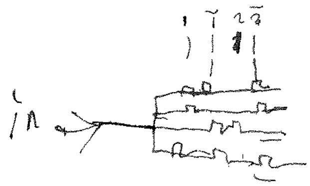
20000
.2 pF

600

Jim Cherry

#16

1 to 4



1 →

Caps on Bit line ~~B~~

24 bit 3-transistor RAM

per each cell		cap	r	
metal cap	$3 \times 24 \mu^2 = 72 \mu^2$	$2-5 \times 10^{-5}$ pF/μ^2		3.6×10^{-3} pF
diff sidewall	$6 \times 4 \mu = 24 \mu$	20×10^{-5} pF/μ		4.8×10^{-3} pF
contact	1 contact	150×10^{-5} pF/contact (guess)		$.15 \times 10^{-3}$ pF
diff field	$36 \mu^2$	7×10^{-5} pF/μ^2		2.5×10^{-3} pF

~~1.105~~
1.1×10^{-2} pF

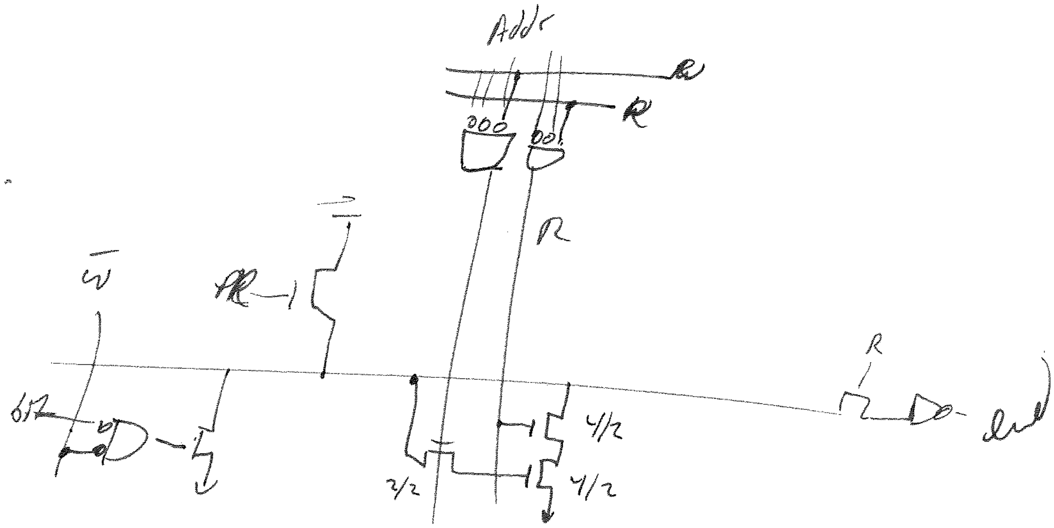
Total cap on a transistor
.26 pF

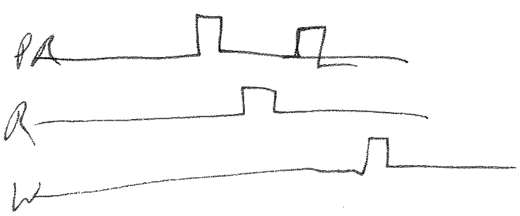
10kΩ/μ^2

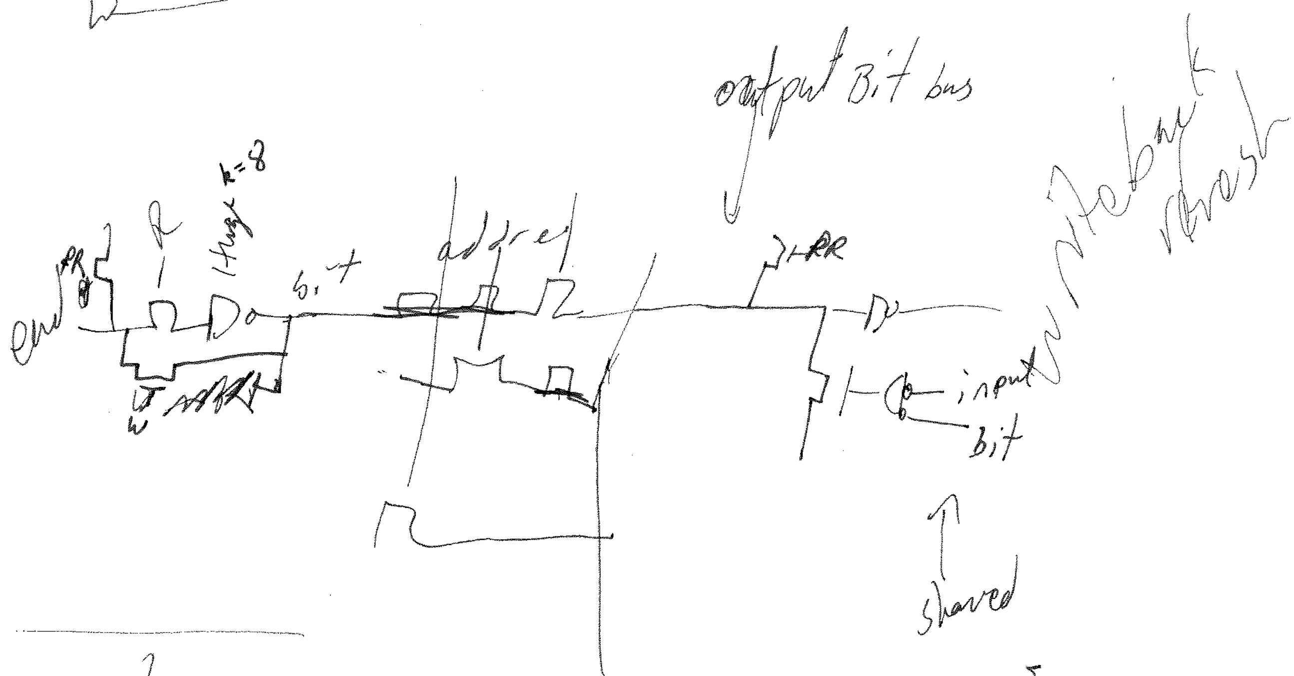
15800 cycle/sec

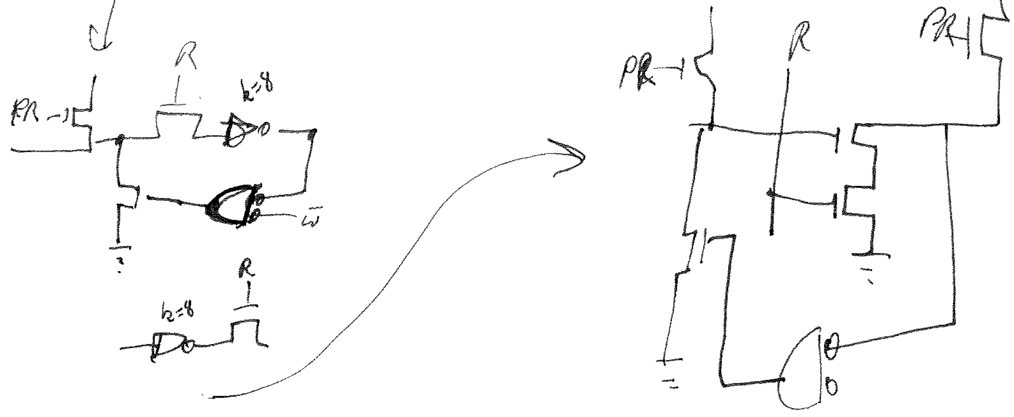
1.5 k

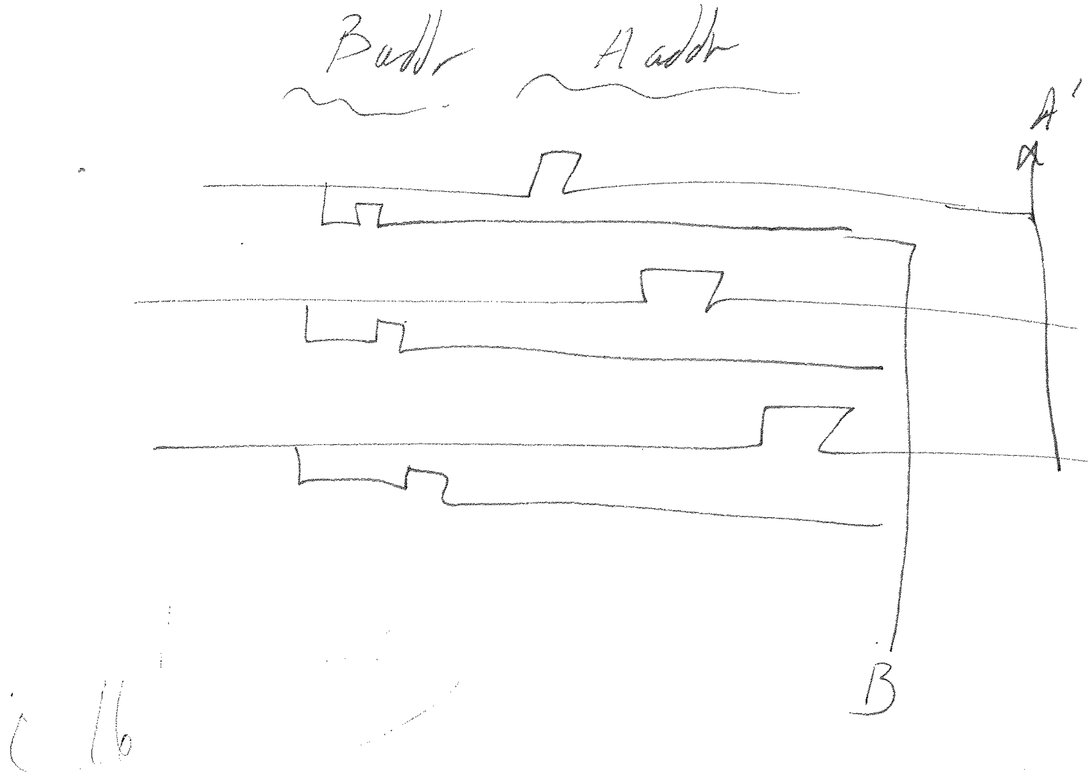
3 mk


Addr
R
R
$\bar{w}$
PR
R
4/2
4/2
2/2
4/2

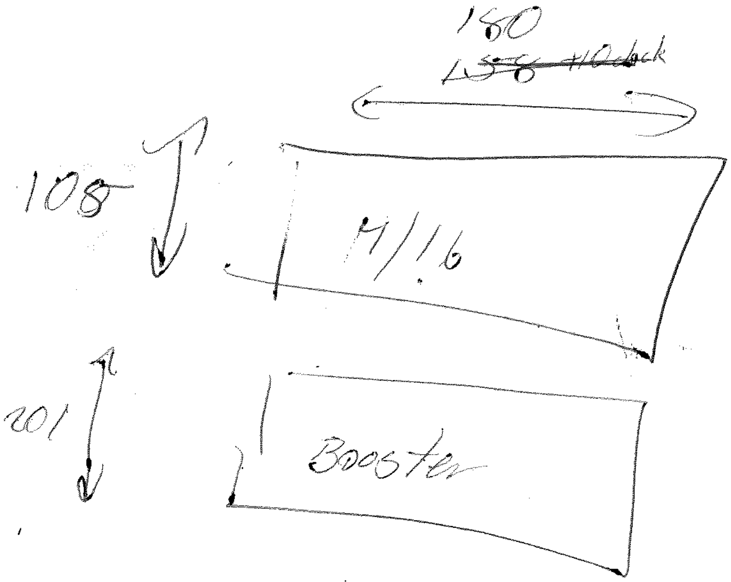

PR
R
W

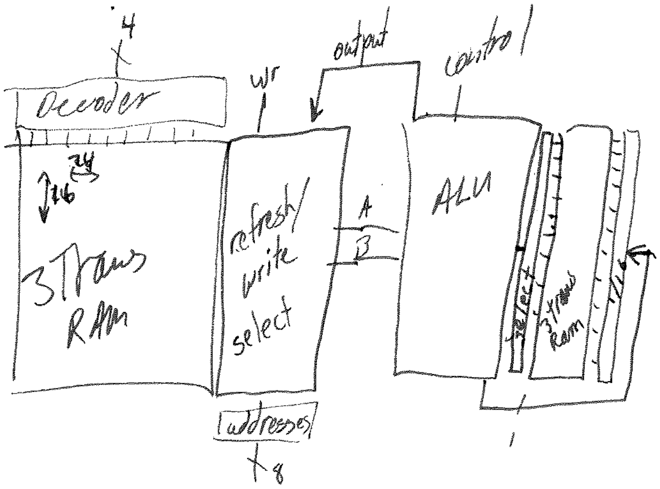

output Bit bus
k=8
R
Huge
bit
address
PR
input
bit
shared
writeback
refresh
w


PR
R
k=8
$\bar{w}$
k=8
R
PR
R
PR

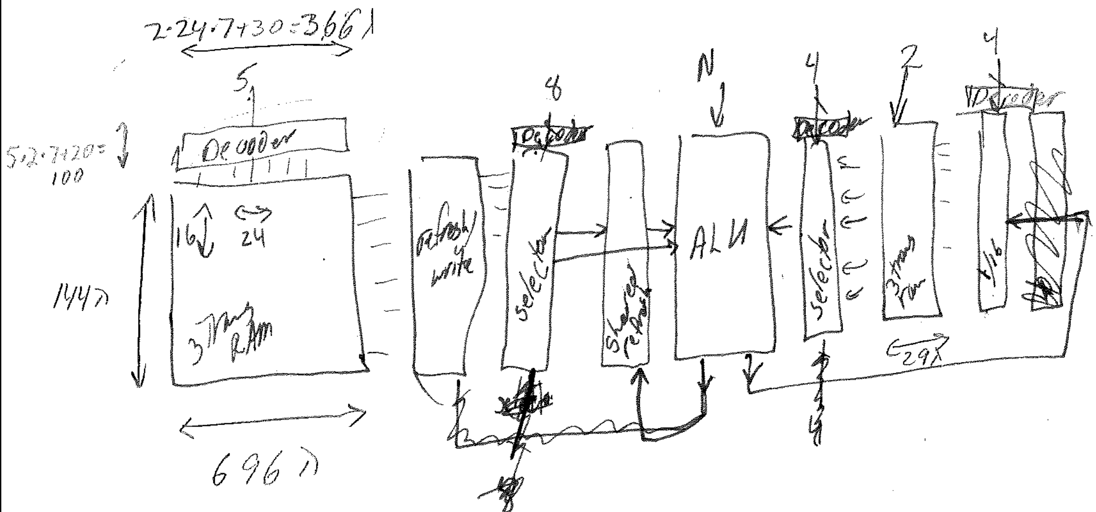


5/24

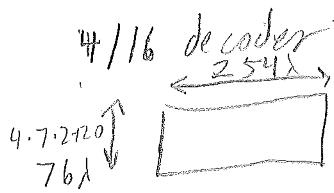




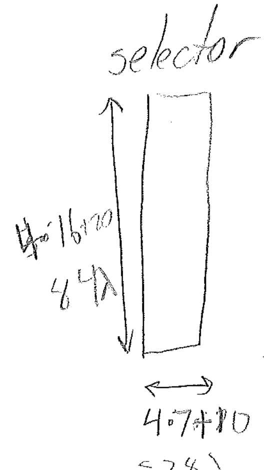
4
Decoder
wr
output
control
16
24
3Trans RAM
refresh/ write select
A
B
ALU
select
3 trans Ram
1/16
addresses
8
1



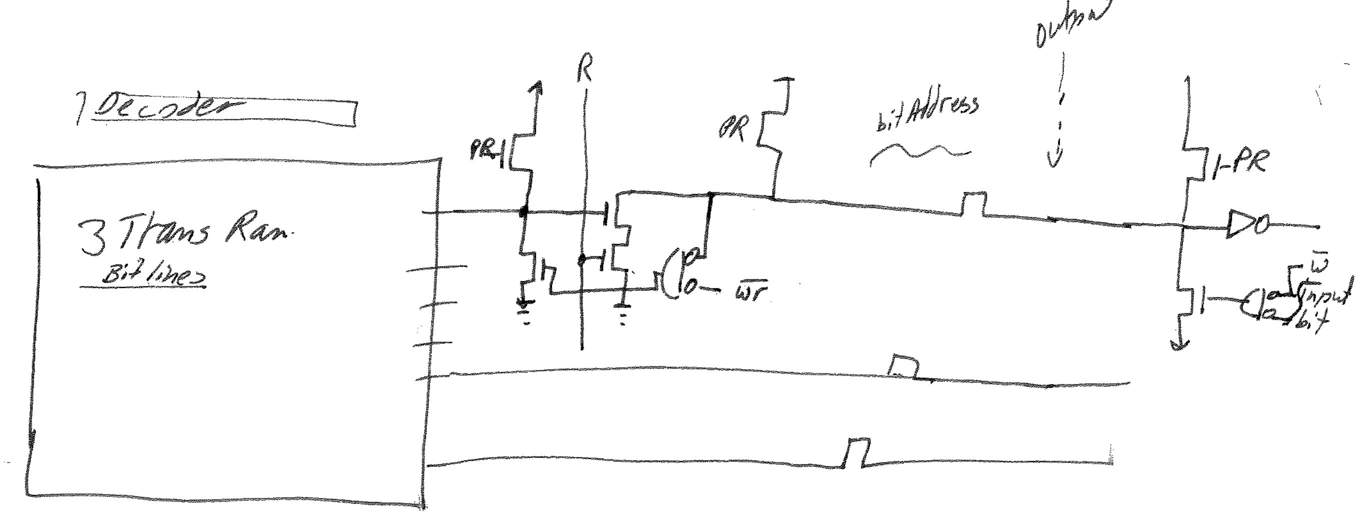
2·24·7+30=366λ
5
5·2·7+20= 100
Decoder
16
24
144λ
3Trans RAM
696λ
refresh/ write
8
selector
shared refresh
N
ALU
4
Decoder
selector
2
3 trans Ram
29λ
4
Decoder
1/16



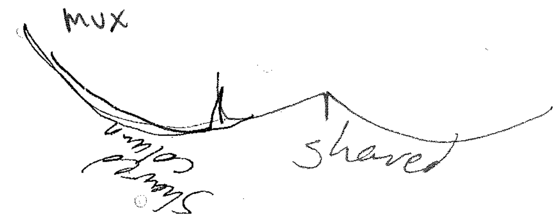
4/16 decoder
254λ
4·7·2+20
76λ



selector
4·16+20
44λ
4·7+10
=38λ



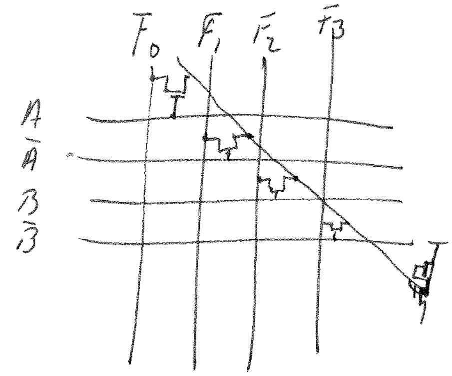
1 Decoder
3 Trans Ram
Bit lines
PR
R
ØR
bit Address
output
1-PR
wr
w̄
input bit



mux
shared column
shared

James Cheng

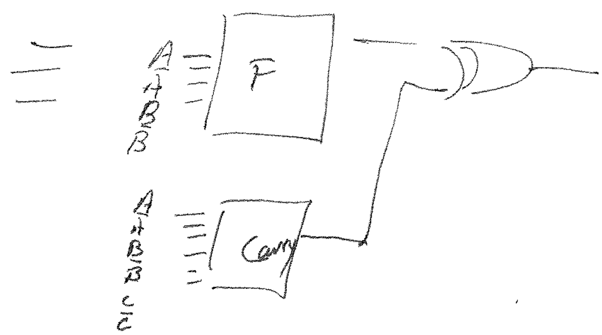
ALU

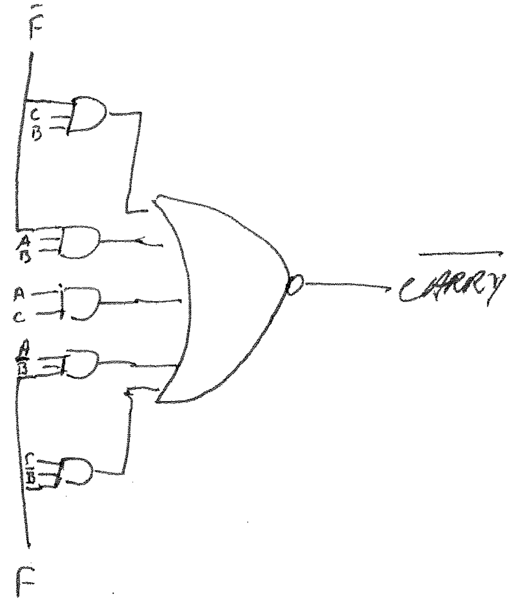


$\equiv F$

$\bar{A} \cdot \bar{B} = A$

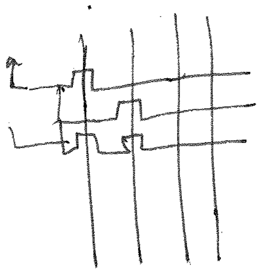
F_0	F_1	F_2	F_3	out	
0	0	0	0		0
0	0	0	1	~~$\bar{B}$~~	0
0	0	1	0	B	0
0	0	1	1	T	0
0	1	0	0	$\bar{A}$	0
0	1	0	1		A+
0	1	1	0		
0	1	1	1		
1	0	0	0		0
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	φ	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

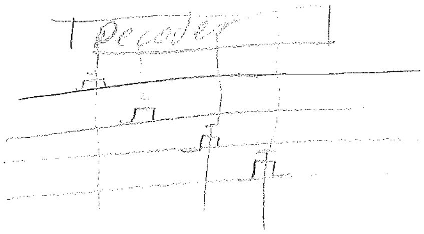


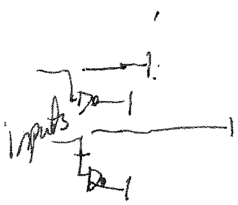


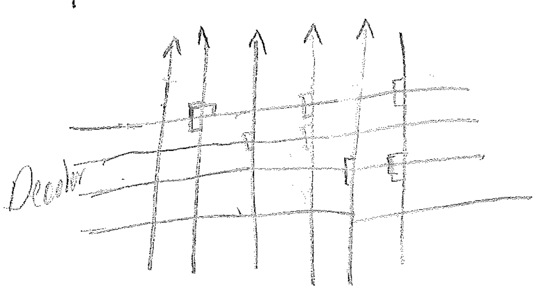
selector

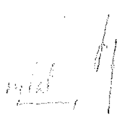
Decoder

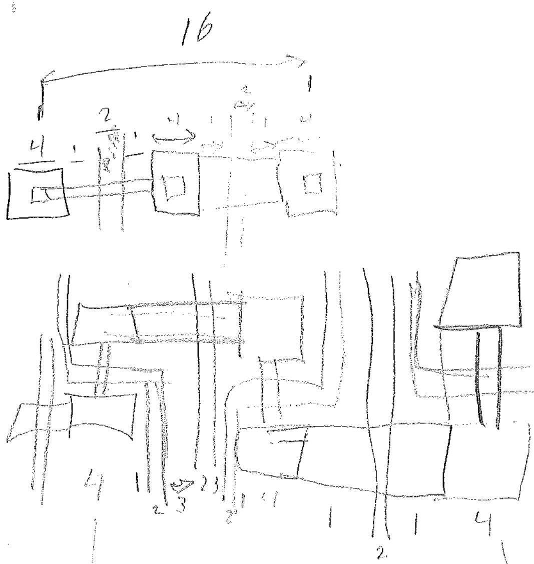












$26/2 = 13\lambda$

need

5/24 decoder

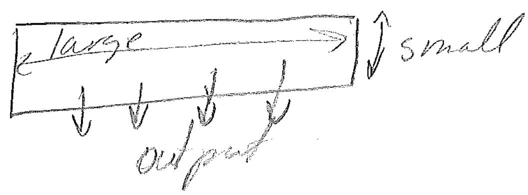
binary word

1 bit high rest low

3 of 4/16 decoder

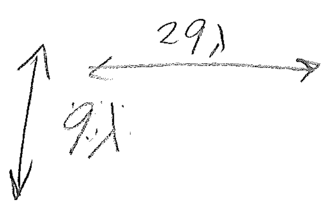
must be as flat as possible

100λ



24·7+30

<u>ram cell</u> (6λ storage)



Area = 261 λ^2

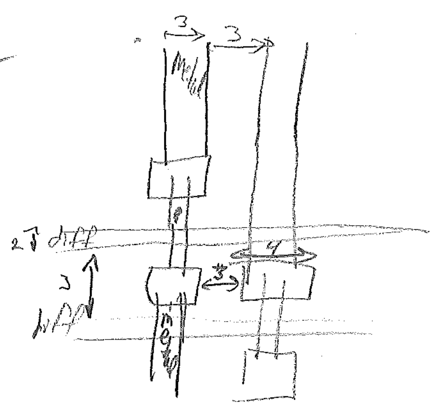
Total size (1^{st} approx)

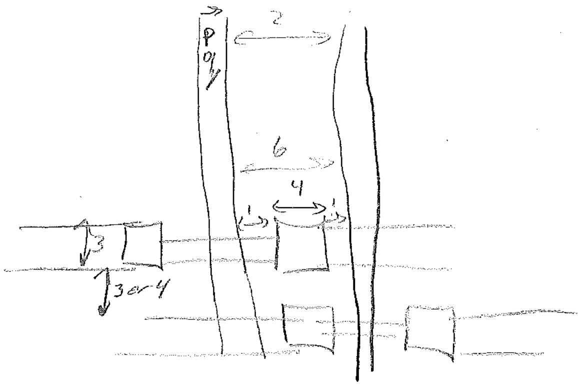
height = 250λ

width = 696 + 30 + 254·3 = 1488

figuring decoders end to end

selector





if metal ↓ diff →

↔ 7λ/per selection line
possibly 6λ/per

↕ 5λ/per selected lines

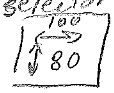
If metal → poly ↓

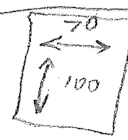
↔ 4λ/per + 10

↕ 7λ/per selected line

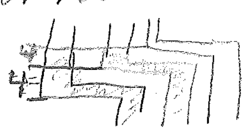
slower (long poly path)

16 bit selector





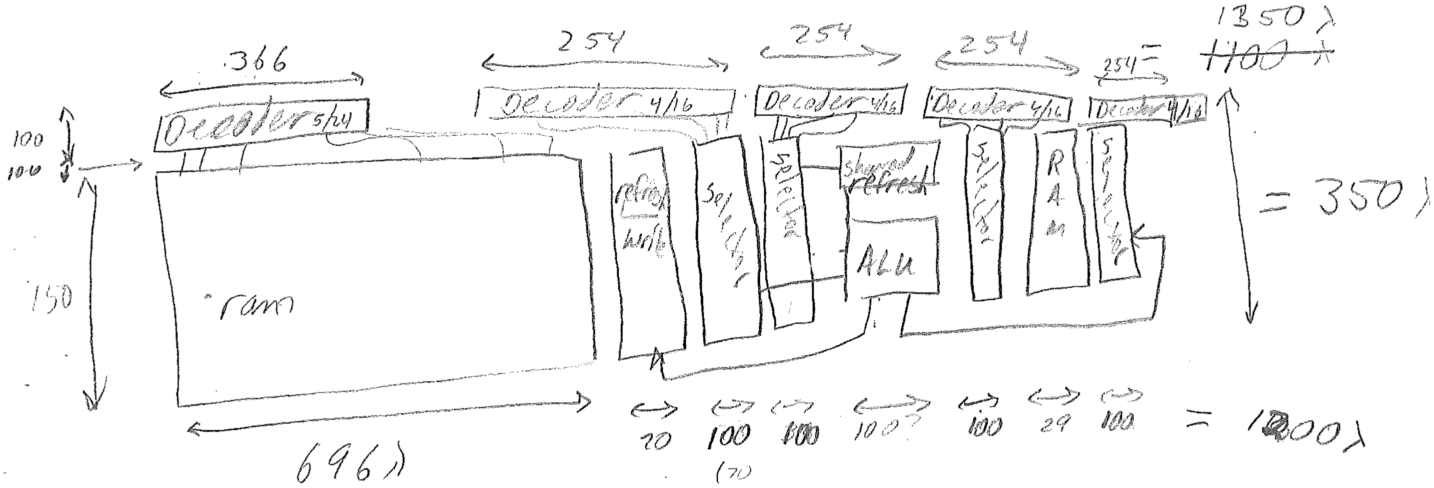
River route



↕ : *spacing* + *width*

metal = 6
diff = 5
poly = 4

if mix poly + metal
maybe 2.5/per line



Area = 1350λ · 350λ = 472 000 λ²

1 cm × 1 cm max size of chip

$500 k \lambda^2 \cdot (2 \mu m/\lambda)^2 = 2000 k (\mu m)^2$

$\mu m = 10^{-6} m = 10^{-4} cm$

$= 2 \times 10^6 (10^{-4} cm)^2$

$= 2 \times 10^{-2} cm^2$

$= .02 cm^2$

put on 50 of these without wiring

Useof Central Proc.

Regist~~ers~~ names

keep 3 address of friends
- tic
- tac
- toe

X

Y

Z

output-to: target

output data: out

messages arrive here
- tic-in
- tac-in
- toe-in

null contains 0

Flag names

condition flags that you can test
- tic-full
- tac-full
- toe-full
- out-full
- a
- b
- c
- true

for global sync. → global
"Ored" with all other globals in machine

false

When message arrives
goes into {tic-in, t -in, t -in} + sets tic-full

to send: put data in out
put address in
out-full is set

Code for symulator 2/23

```
'inst= (Add <cd-#> A B S D)

       (move  2  A B)

  (Assemble  'inst)   (assemble '(prog (Add  )
                                        (    )
                                        (    )))
  (Run (Assemble 'inst))

  (repeat  I  1 23
         ( Add I ..))
```

Does with I = 1, 2 ... 23

$AC \cup BC \cup AB$

$A \cup B \cancel{\cup AB}$

Bool A B → A

Bool A F → A

ADD A B → A

SUB A B → A

AND (lenth A) → F   which is And A $\cancel{B}$ → F   x2Y

OR (lenth A) → F   which is Or A F → F   x2Y

compare (length A, len B) → F   which is $A \cdot B \cdot F + \bar{A}\bar{B} \cdot F \rightarrow F$ x2Y

F → A

A → F

B → A

F → F

F → A , A → F

inc A → A

dec A → A

permute A → A   hack by calling →(A,A); good for shift

0 → A

1 → A

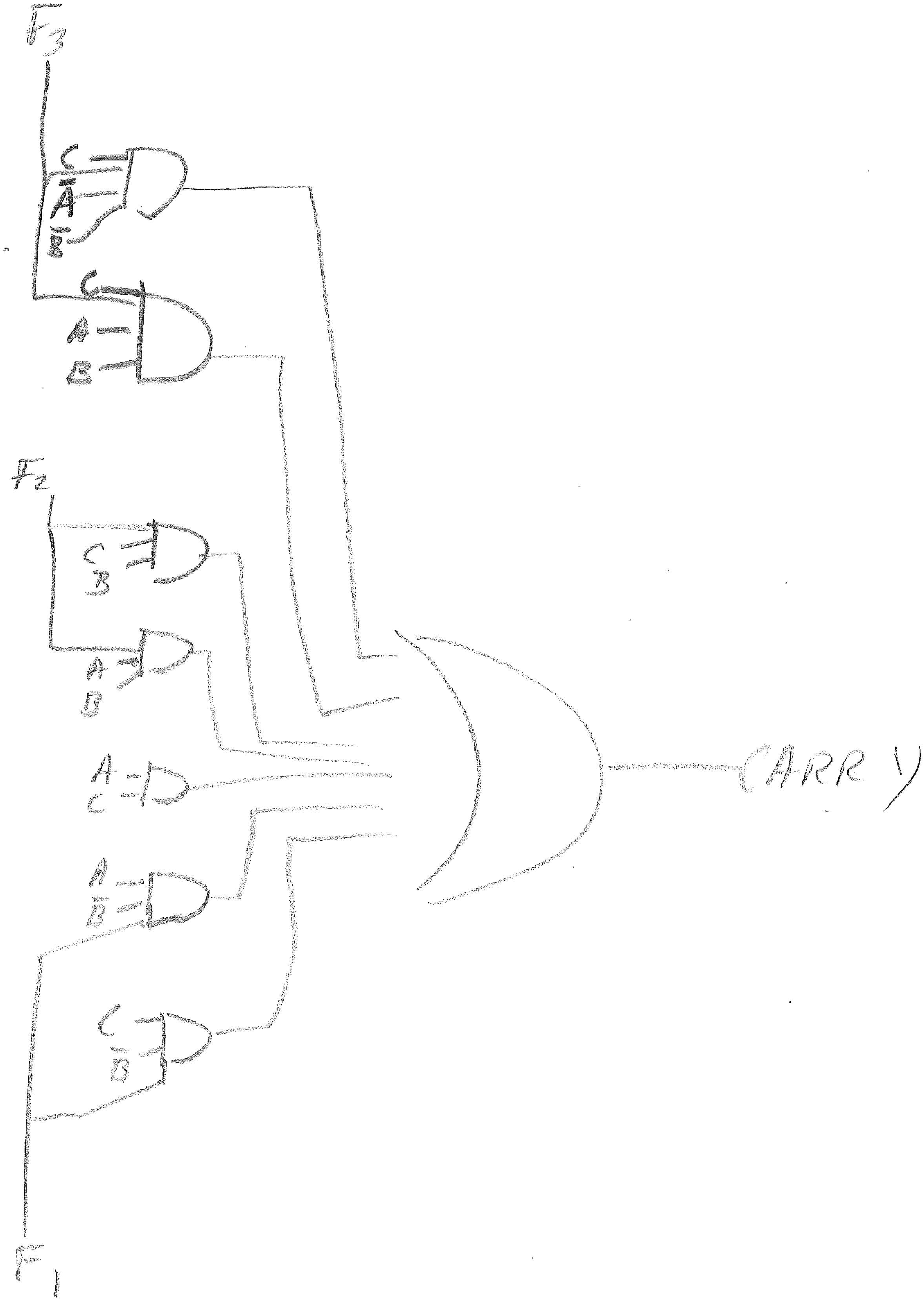
0 → F

1 → F

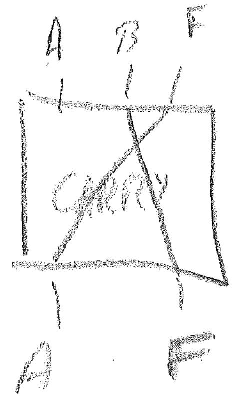
Possible want: shift-move (length A) → B

$EQ = \overline{XOR}$

F_1	F_2	F_3	
0	0	1	$C \cdot (EQ\ A\ B)$ compare
0	1	0	$AC + A\bar{B} + \bar{B}C = MAJ(A\bar{B}C)$ subtra
1	0	0	$AC + AB + BC = MAJ(ABC)$ Add
0	0	0	AC
1	1	0	$A + C$


F_3
C
$\bar{A}$
$\bar{B}$
C
A
B
F_2
C
B
A
B
A
C
A
$\bar{B}$
C
$\bar{B}$
F_1
CARRY

F_4 $\quad$ F $\quad$ A

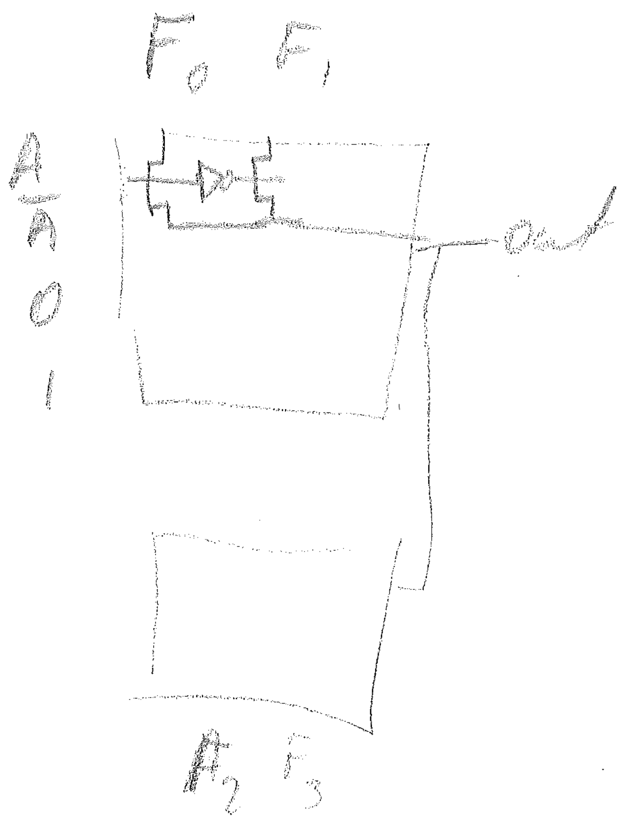


$F_1 = \text{Maj}(A\,B\,F)$ $\quad$ F

$F_2 = \text{Maj}(A\,\bar{B}\,F)$ $\quad$ F

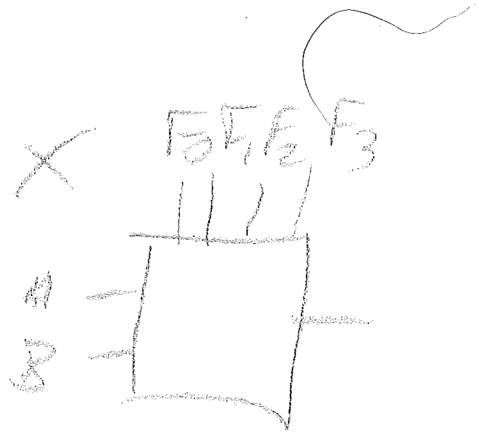
$F_3 = \text{Comp}(A\,B)$ $\quad$ 0

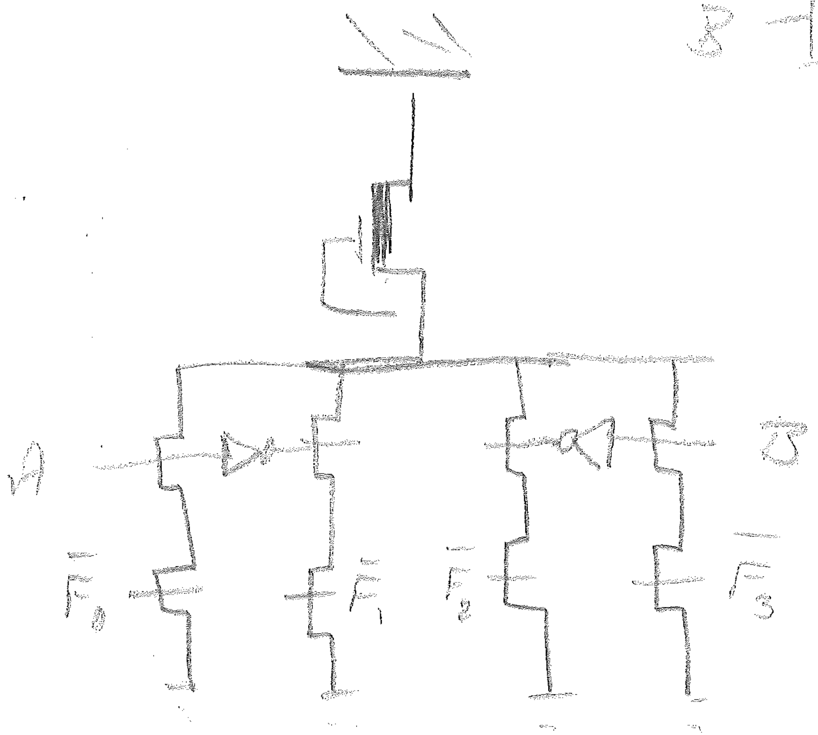
$F_4 \stackrel{?}{=} \text{shiftmove}(A\,B)$ $\quad$ F



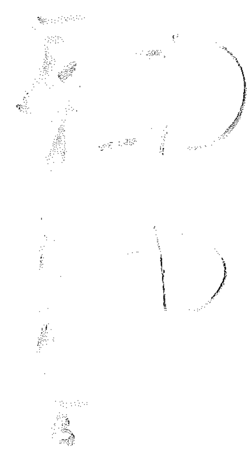
F_0	F_1	
0	0	0
1	1	1
1	0	A
0	1	$\bar{A}$

Function Box



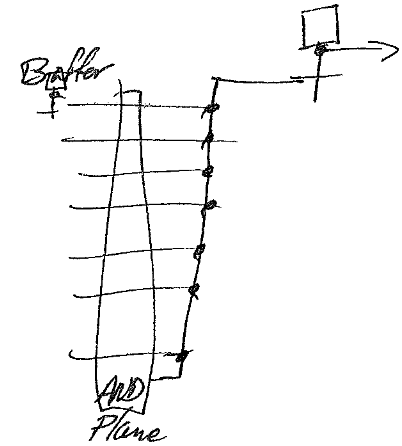


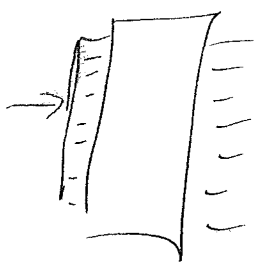
$$A\cdot\bar{F_0} + \bar{A}\cdot\bar{F_1} + \bar{B}\cdot\bar{F_2} + B\cdot\bar{F_3}$$

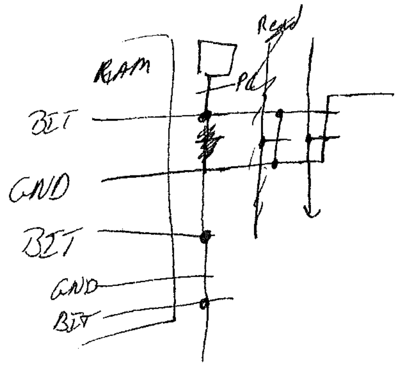


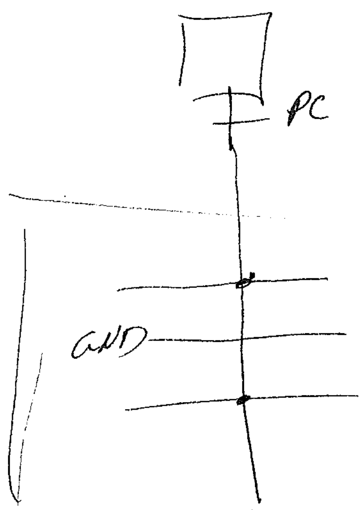
selector 16/1

idea:





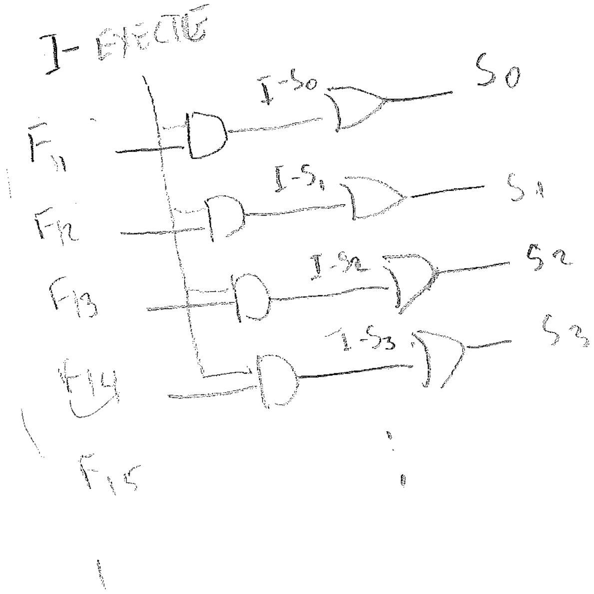


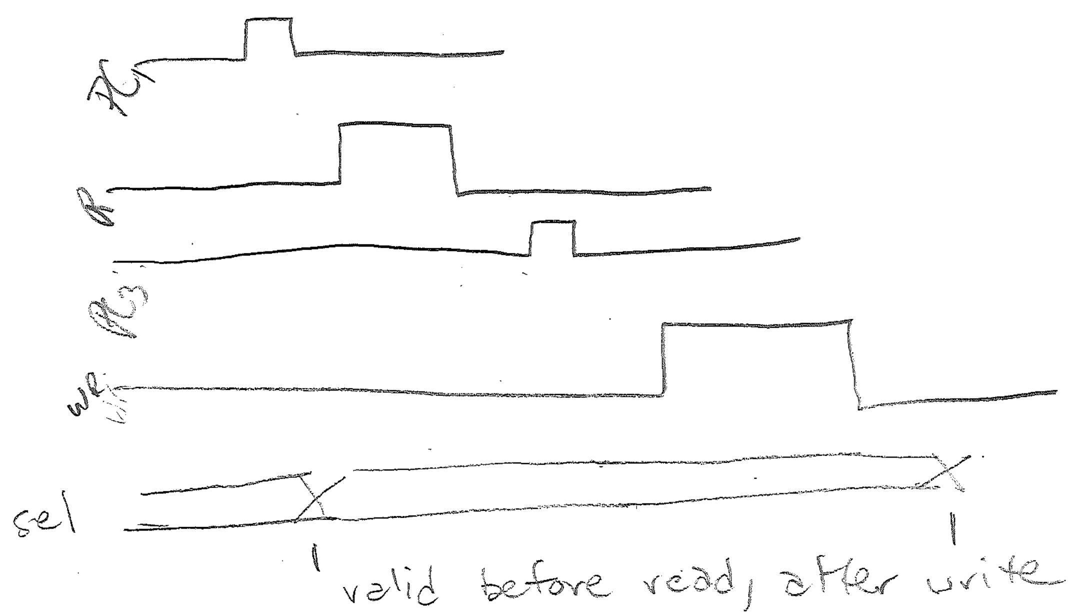


Execute Instruction

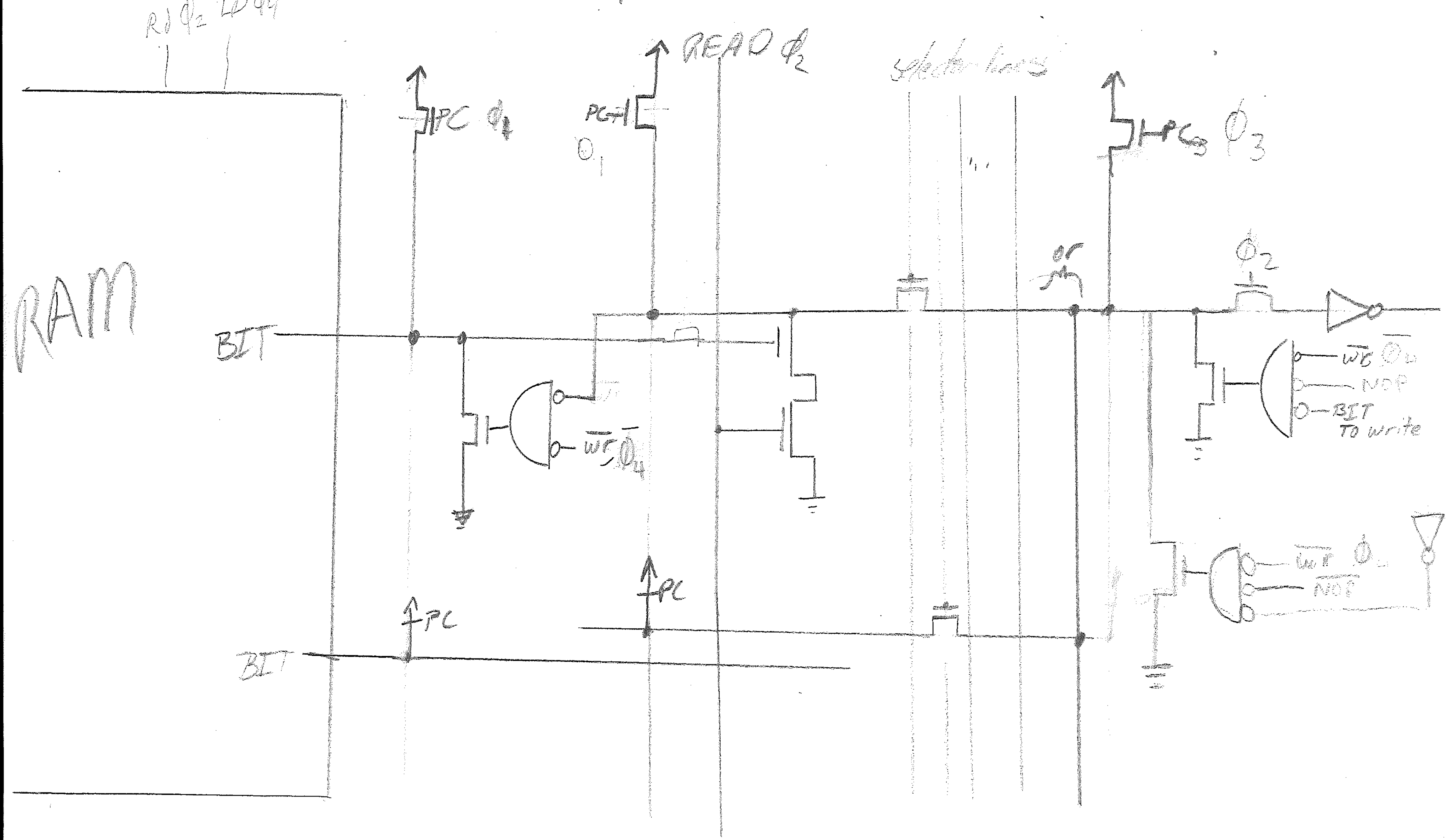
(way to individually program an ALU, take instruction from ~~they~~

one per cell



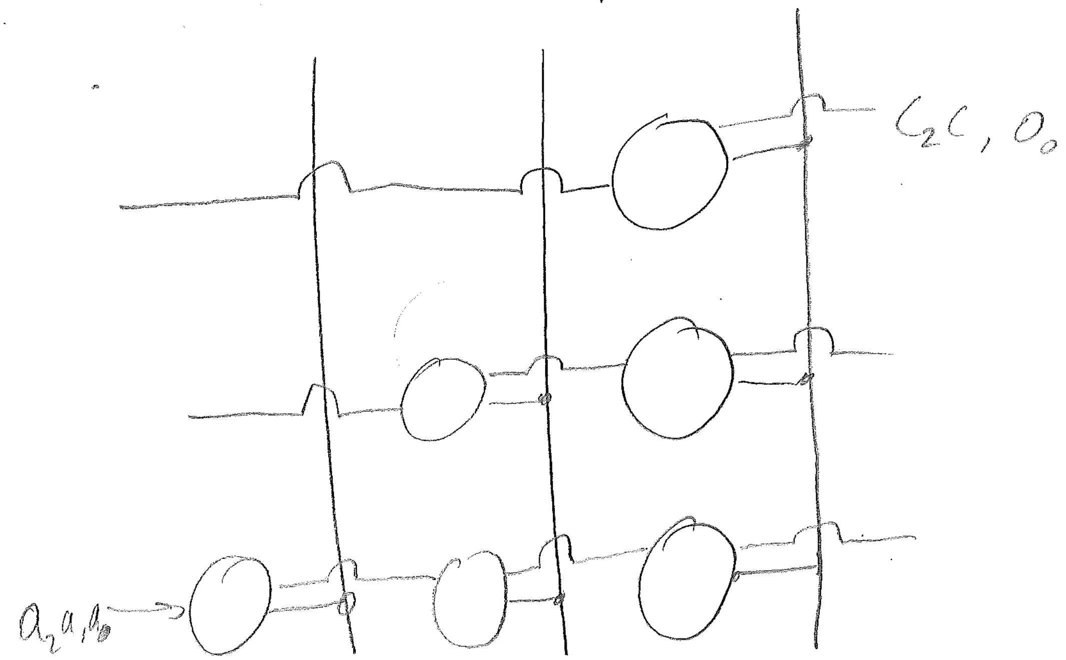

R
sel
valid before read, after write

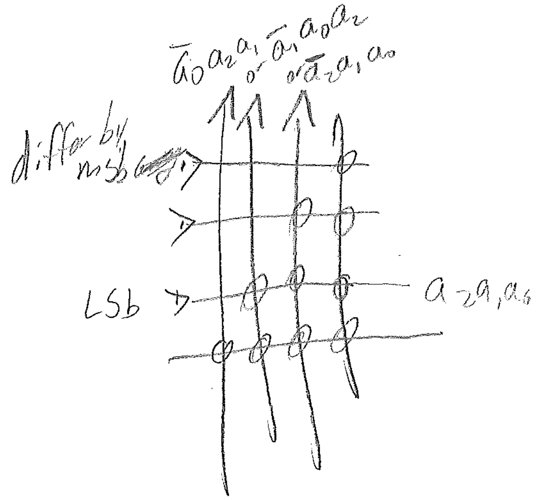
CHERRY-RAM®

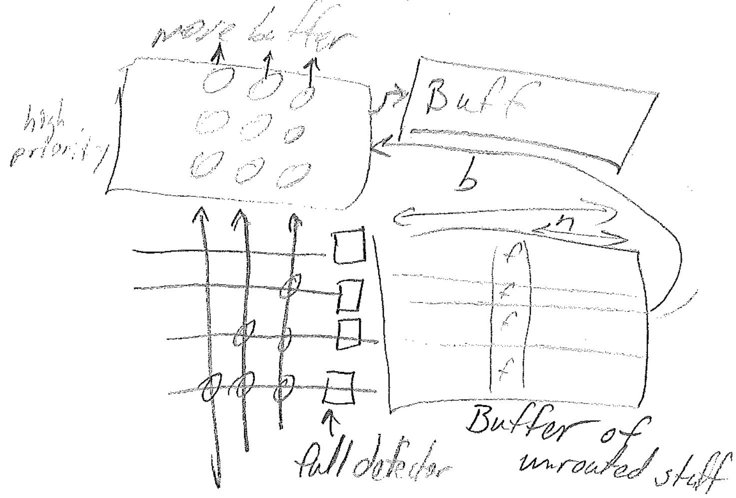


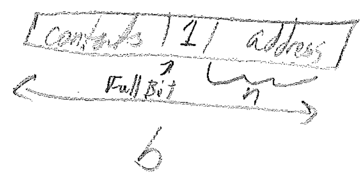
4 phase, non-overlapping clock
Read on Φ2, WE Φ4

Heart (1 per chip)

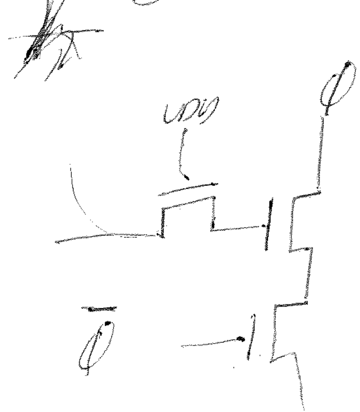


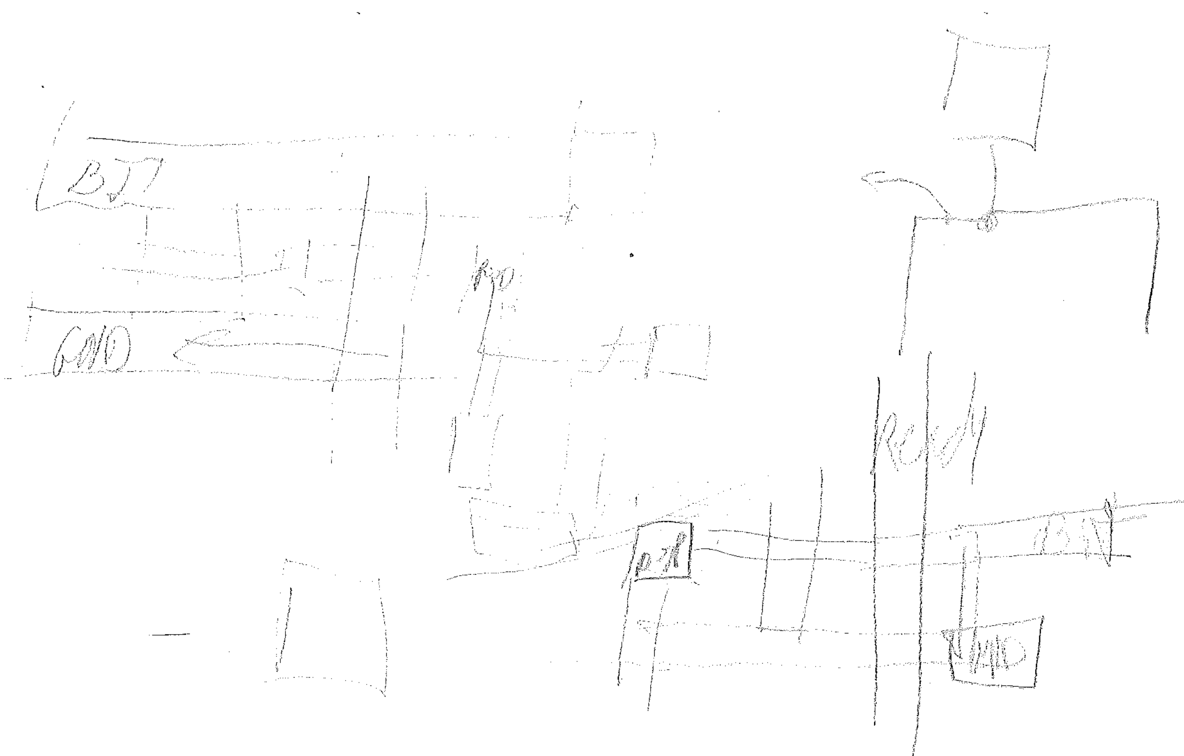


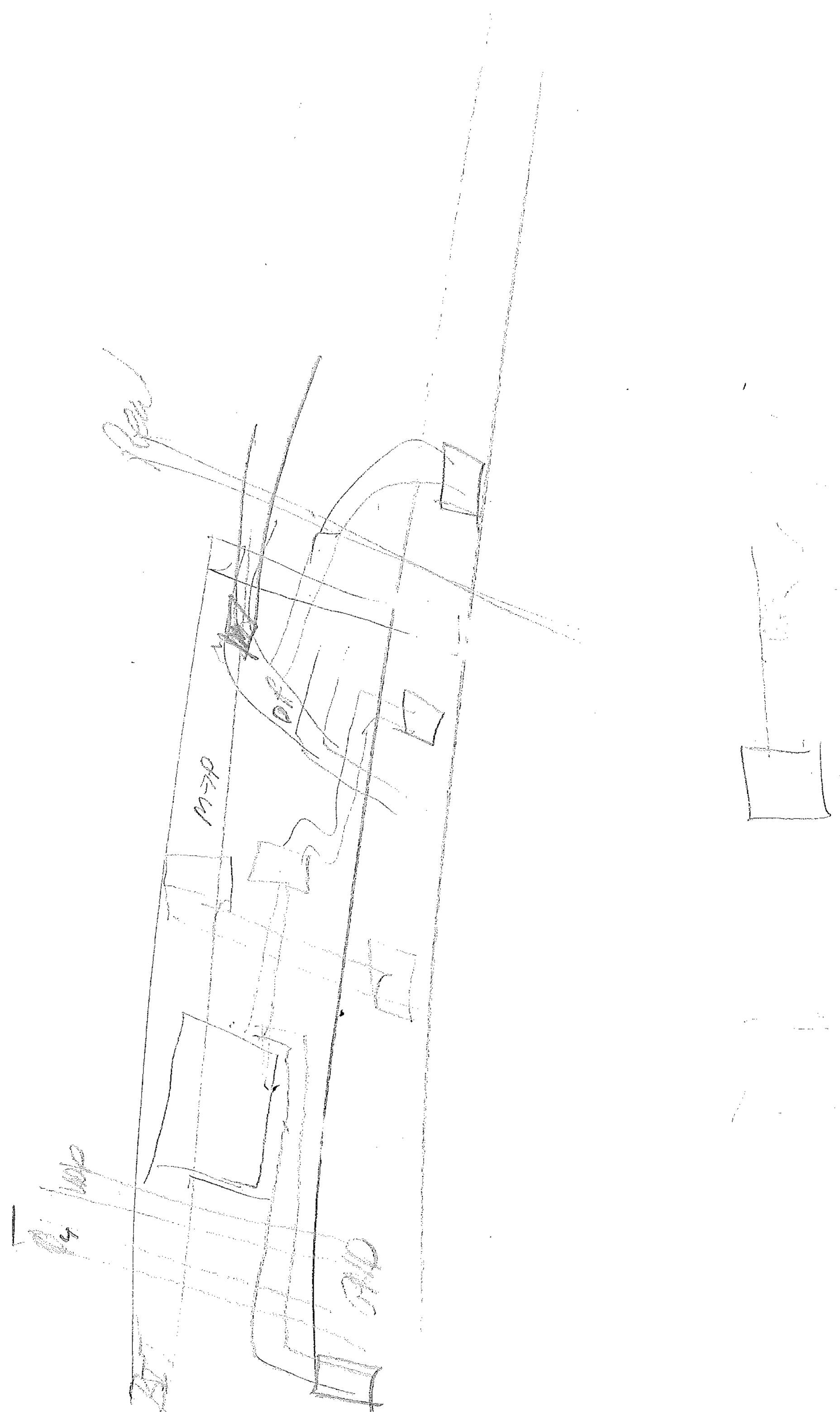




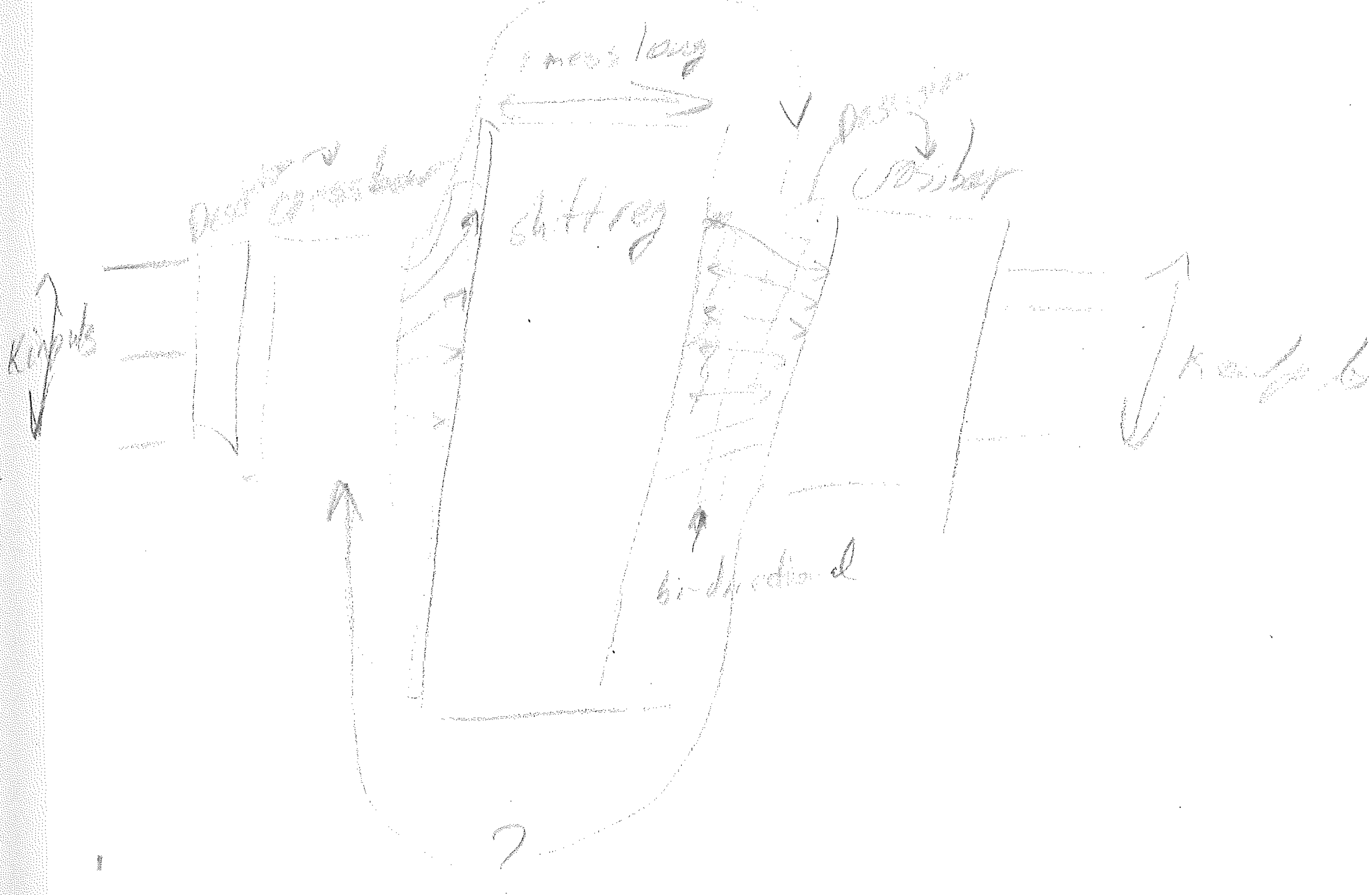
Bootstrap driver







Large Dynamically allocated ~~general~~ register system



size:

sh reg 50 bits × # of mess lcg

crossbar input

K inputs

1 + delay of sh reg

1 or K bypass

use:
transparent: bypass string
problems: would need K bypass paths

long string system
tie each string output to input of next
(assuming bidirectional registers — or alternating registers)
use only every n^{th} input

~~[illegible]~~ Also use long string idea but can take off the output from anywhere (the head of string
prob: using alternating scheme, can only take odd # messages

prob with many gens; don't know where it is to find next gen
where go / ~~[illegible]~~ unnecessarily
limiting [illegible]

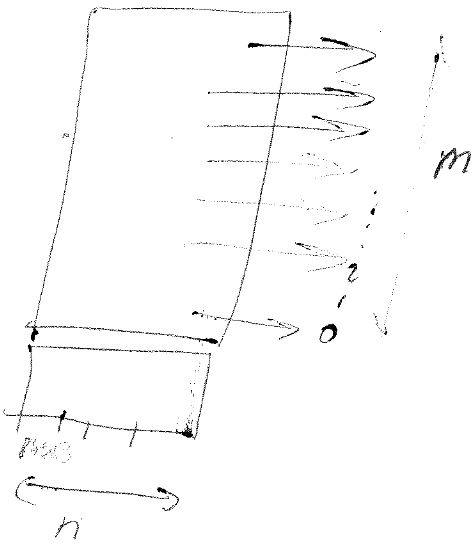
(load "sys>bartuli"

(load "pla>bartuli"

(make-system 'pla)

()k small-selector

To specify Pla for John Bot



need n, List (combination of n inputs to turn on 0th element of m
" " " " " " " 1st " " "

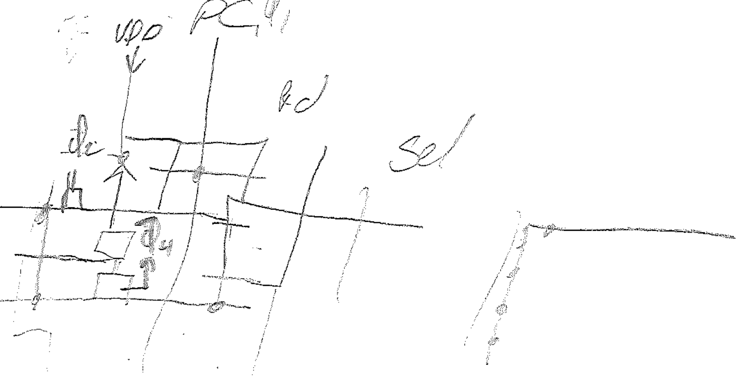
⋮

).

ie if want 0001 to turn on line 0 then
0101 " " " line 1

4, (0001 0101 ...)

RAM



$\phi_1 = PC_1$

$\phi_2 = Read$

$\phi_4 = wr$

$\phi_3 = PC_3$

up-down lines needed

RAM

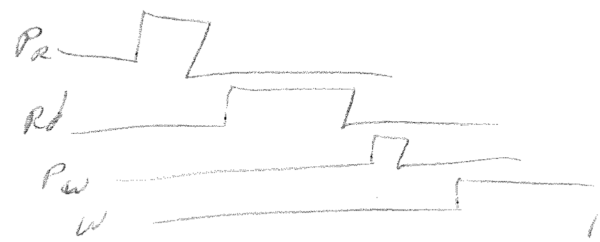
$\phi_2 \quad \phi_4$

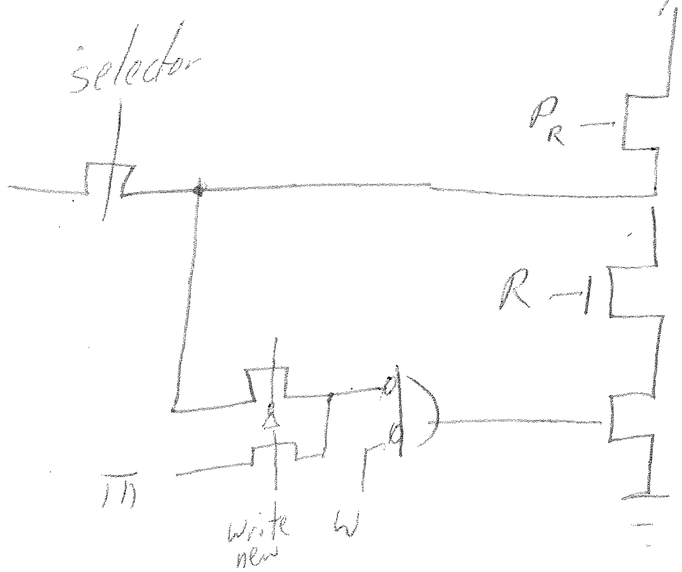
VDD $\phi_1 \bar{\phi}_4 \phi_2$ Selector lines $\cdot \phi_2$

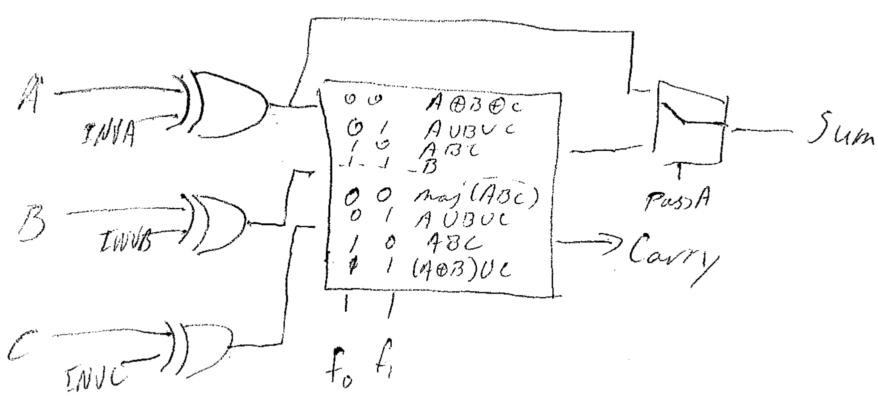
common share

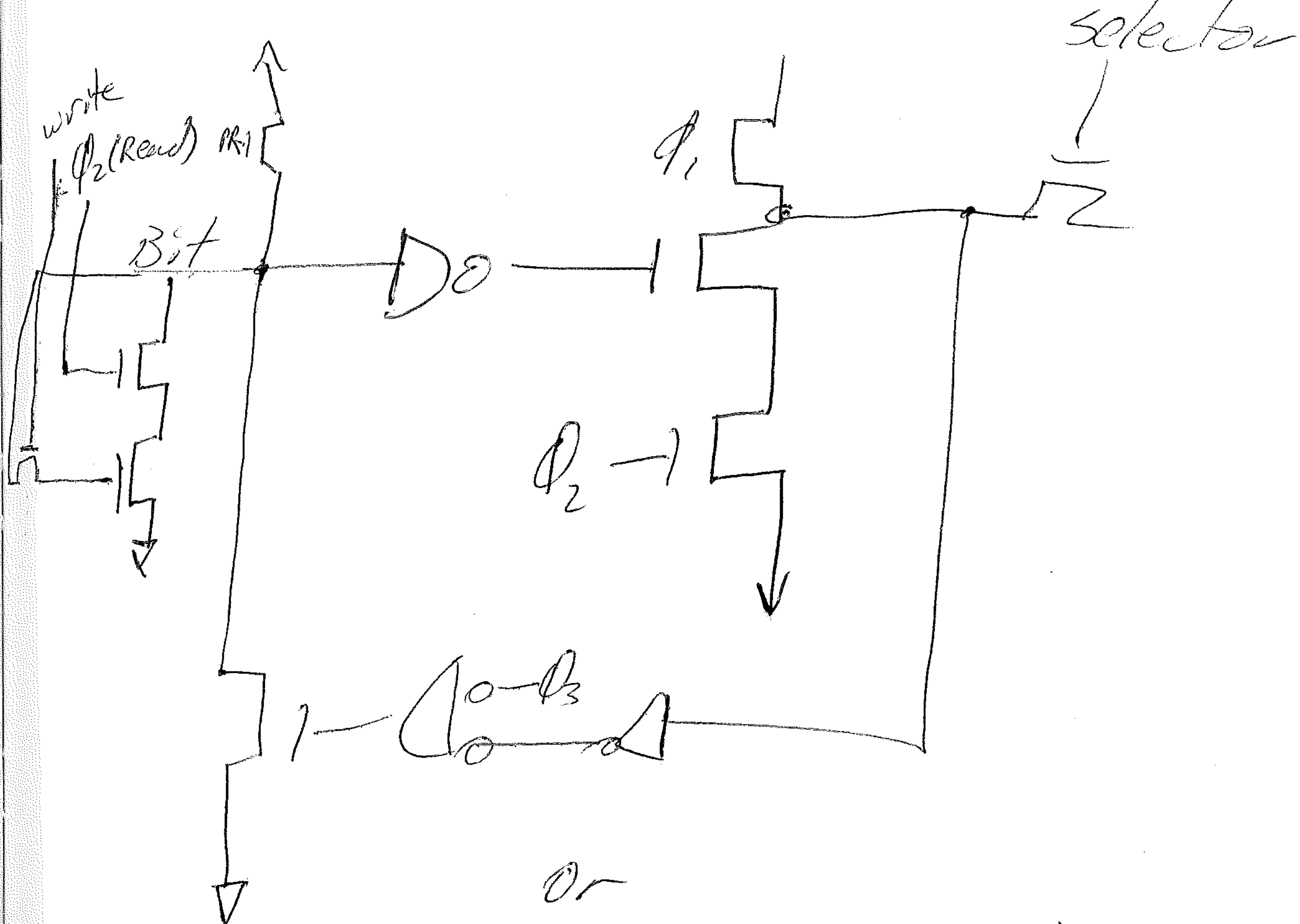
$\phi_3 \quad \phi_2 \quad \overline{\phi_4}$

Flag register

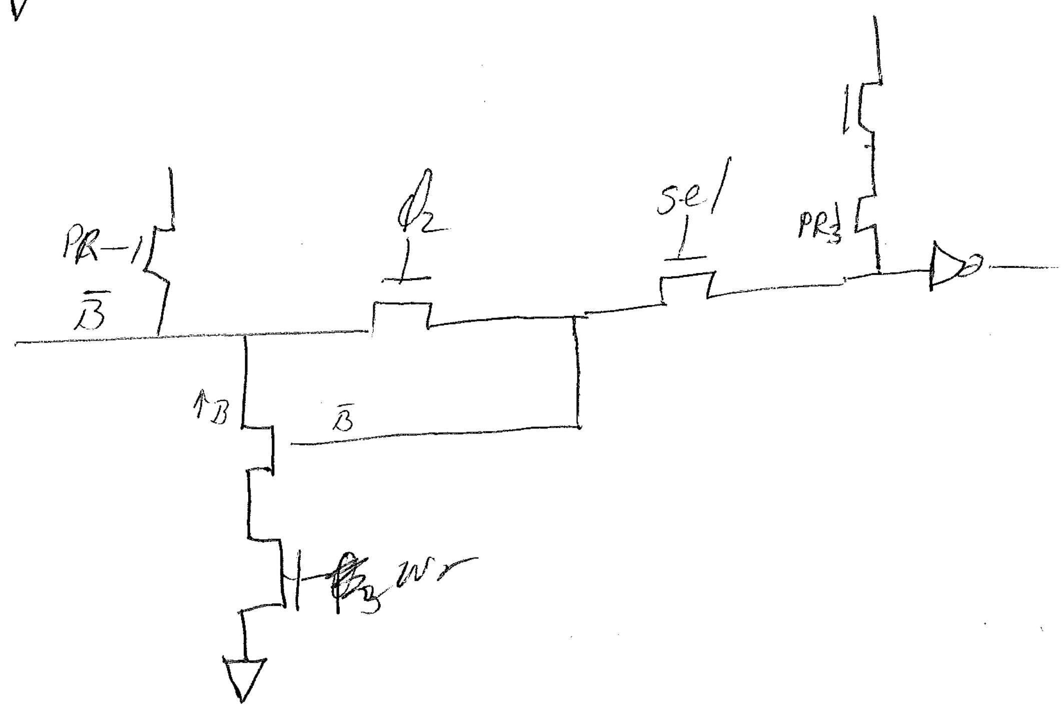


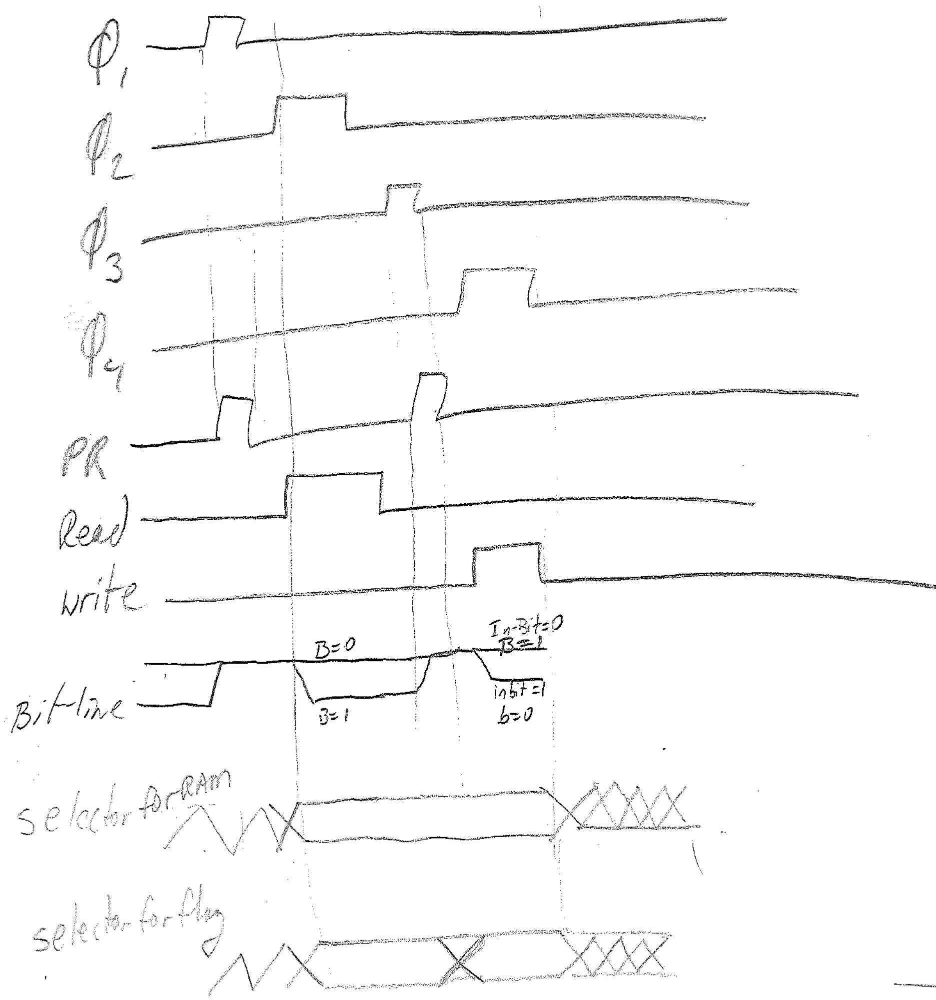





selector
write
ϕ_2(Read)
PR-1
Bit
ϕ_1
ϕ_2
ϕ_3

Or


PR-1
$\bar{B}$
ϕ_2
sel
PR_3
B
$\bar{B}$
Wr

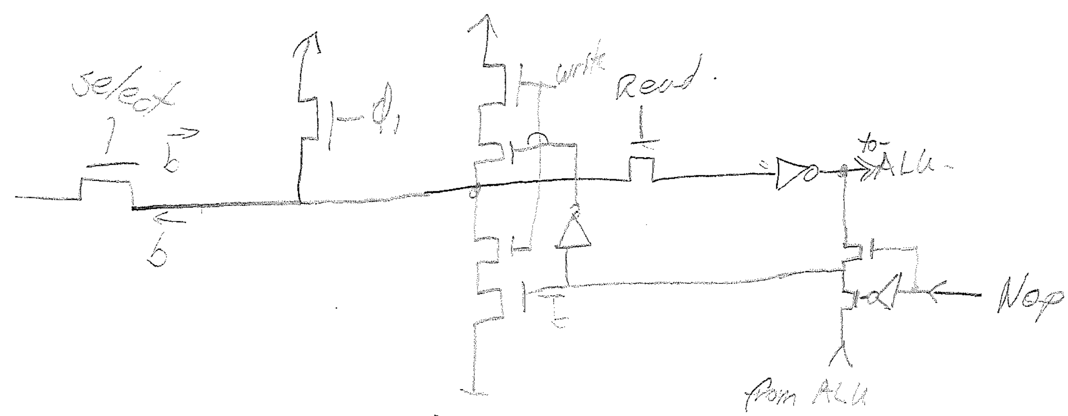


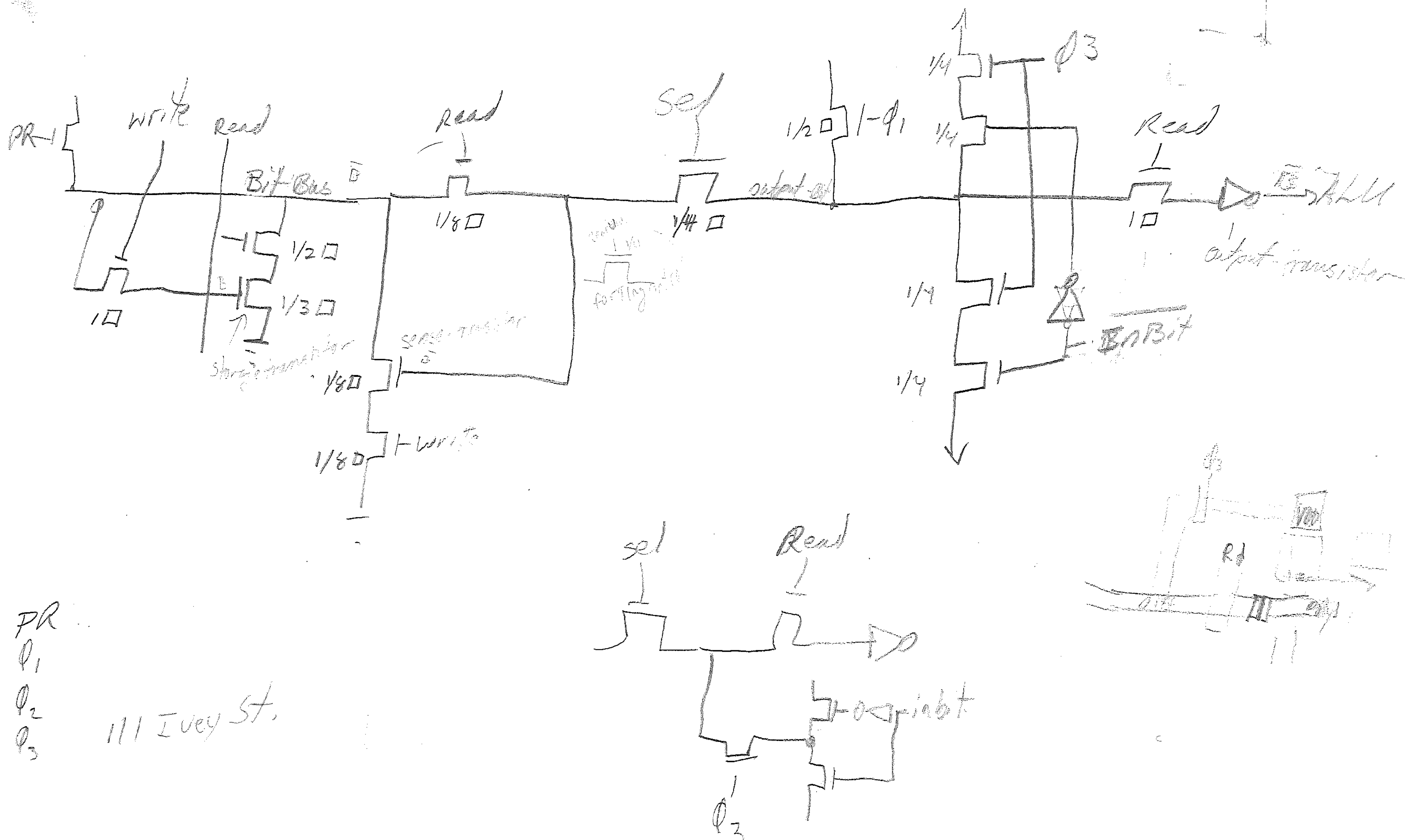
ϕ_1 Precharge before Read (short)

ϕ_2 Read inputs through ALU (long)

ϕ_3 develops the bit to send back
(ALU must be finished before this)
in flag the selector changes state to select bit to write back to.

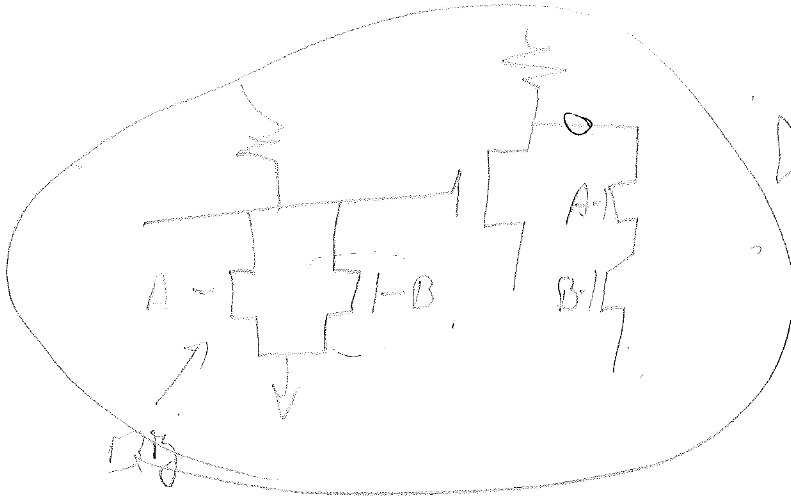
ϕ_4 write bit back into ram





PR
ϕ_1
ϕ_2
ϕ_3

111 Ivey St.



Xor
industry
goodie

xor

A	B	C	A+B+C
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$\Rightarrow \bar{A}\bar{B}\cdot C + \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + ABC$

$\bar{A}BC + \bar{A}\bar{B}C + AB\bar{C} + A\bar{B}C$

A

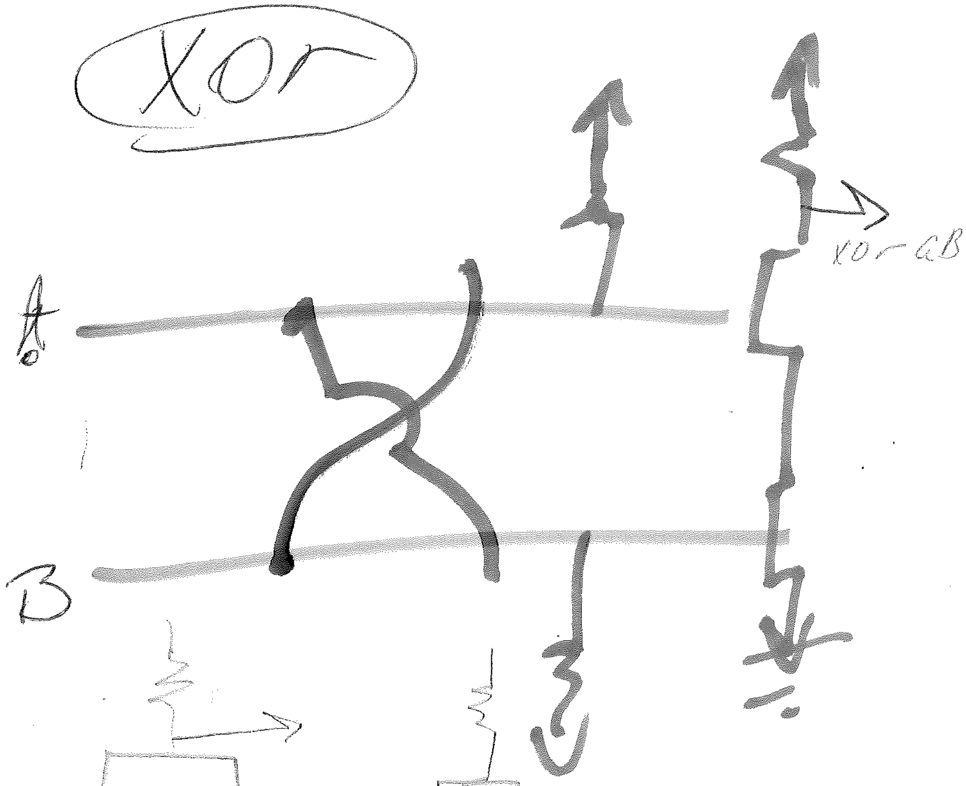
xor

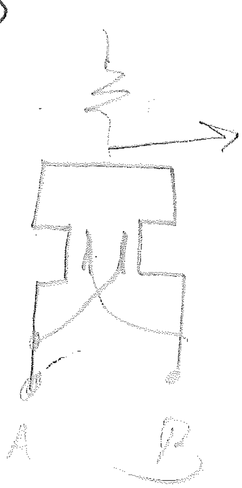
A B' C

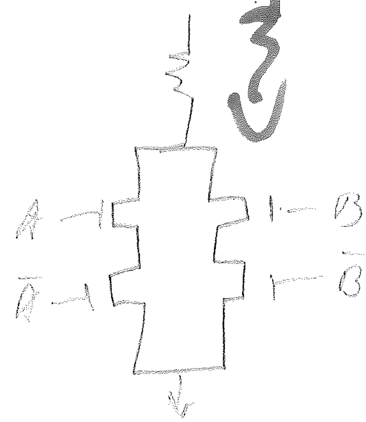
Security

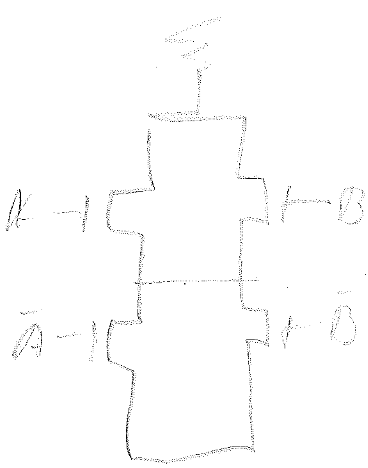
Pla aluplа

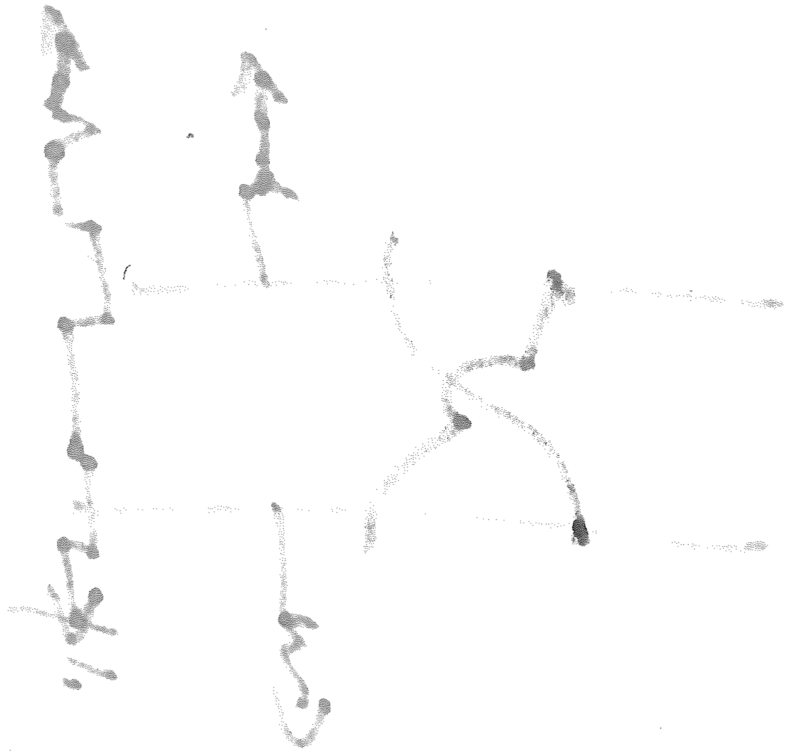
XOr

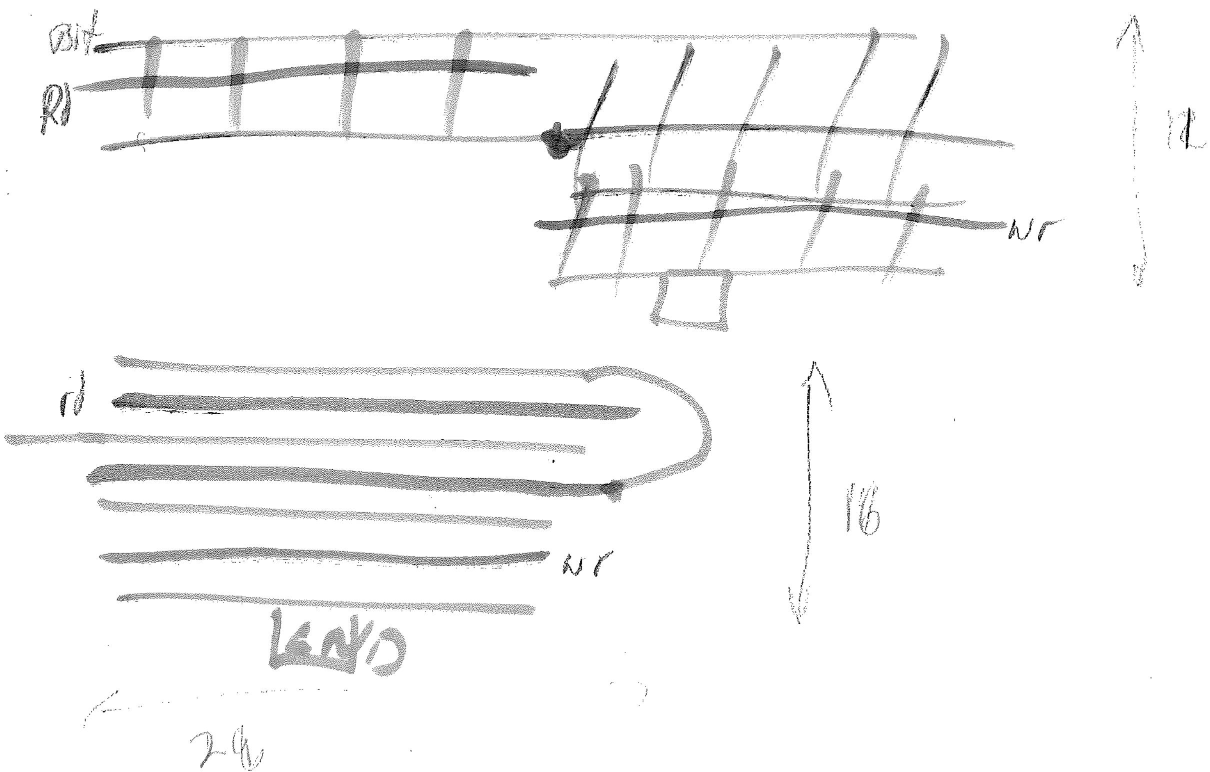




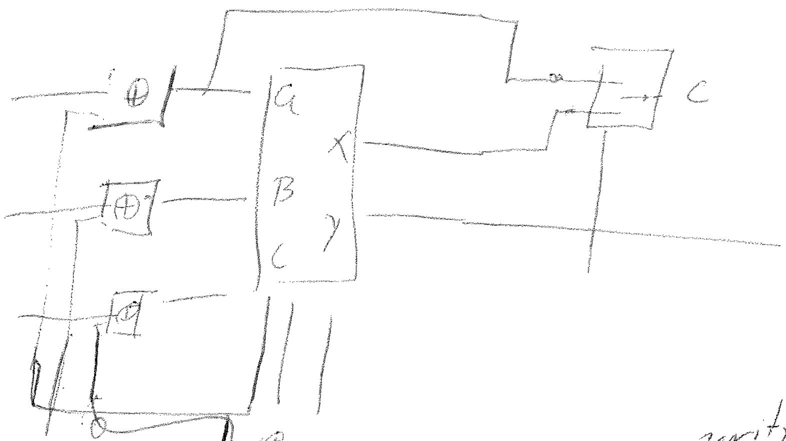








wr
12
16
wr
24



1 ⊕

X 0 0 $x = a \oplus B \oplus C$ = parity $y = maj(a\ b\ c)$

~~X 0 0 $x = \bar{c}$ $y = b$~~

y 0 1 $x = y = a \wedge b \wedge c$

x 1 0 $x + y = a \vee b \vee c$

D 1 1 $x = b$ $y = (a \oplus b) \wedge c$ none

1 1 1 $x = c$ $y = b$

$a \oplus B' = \overline{A \oplus B}$

0	0	0
1	0	1
0	1	1
1	1	0

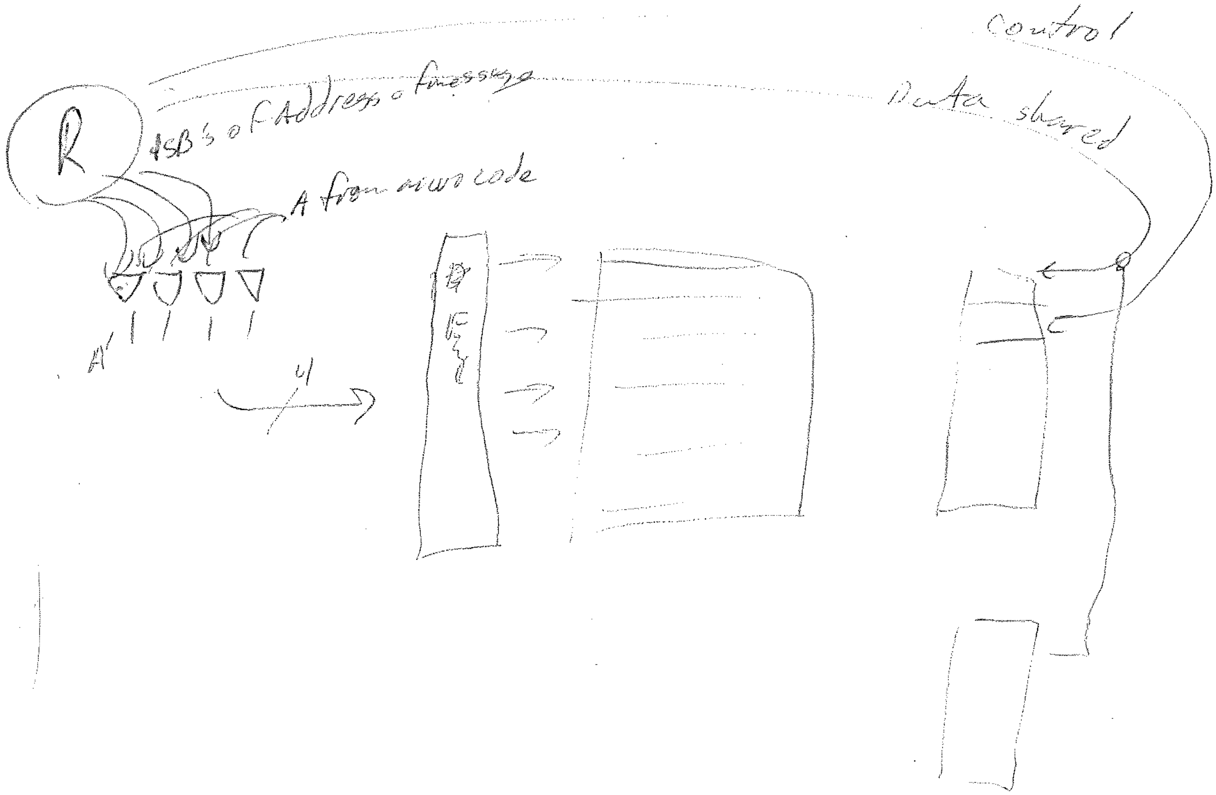
1 1 1

0 = 1

$A \oplus B \oplus C = ABC + \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C}$

$maj(ABC) = ABC + AB\bar{C} + A\bar{B}C + \bar{A}BC$

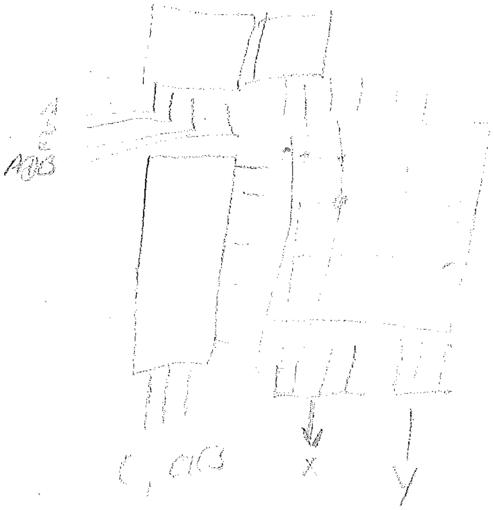
A B C A⊕B



Bit cycles	1 bit info
Message cycle	message delivery rec.
delivery	permutation of all messages in the system

AND plane output

A AB AC
A
B
C
DC
$\bar{D}C$



$\bar{A}B$

$A\bar{C} \vee \bar{A}\bar{B}$

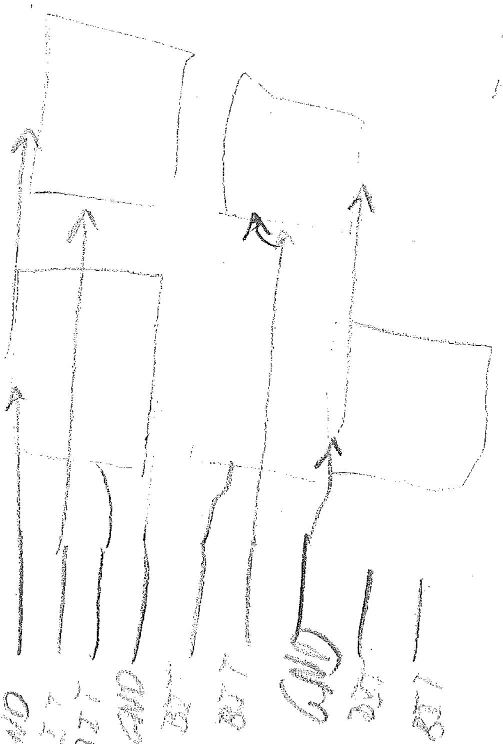
$(A\bar{B} \vee \bar{A}B)\cdot\bar{C} \vee \overline{(\bar{A}\bar{B} \vee AB)}C$

$A\bar{B}\bar{C} \vee \bar{A}B\bar{C} \vee \bar{A}\bar{B} \cdot \bar{A}BC$

$D\oplus C \quad A\bar{B} \vee \bar{A}C$

maj(A,B,C)

$A\bar{B}C + A\bar{B}C + \bar{A}BC$



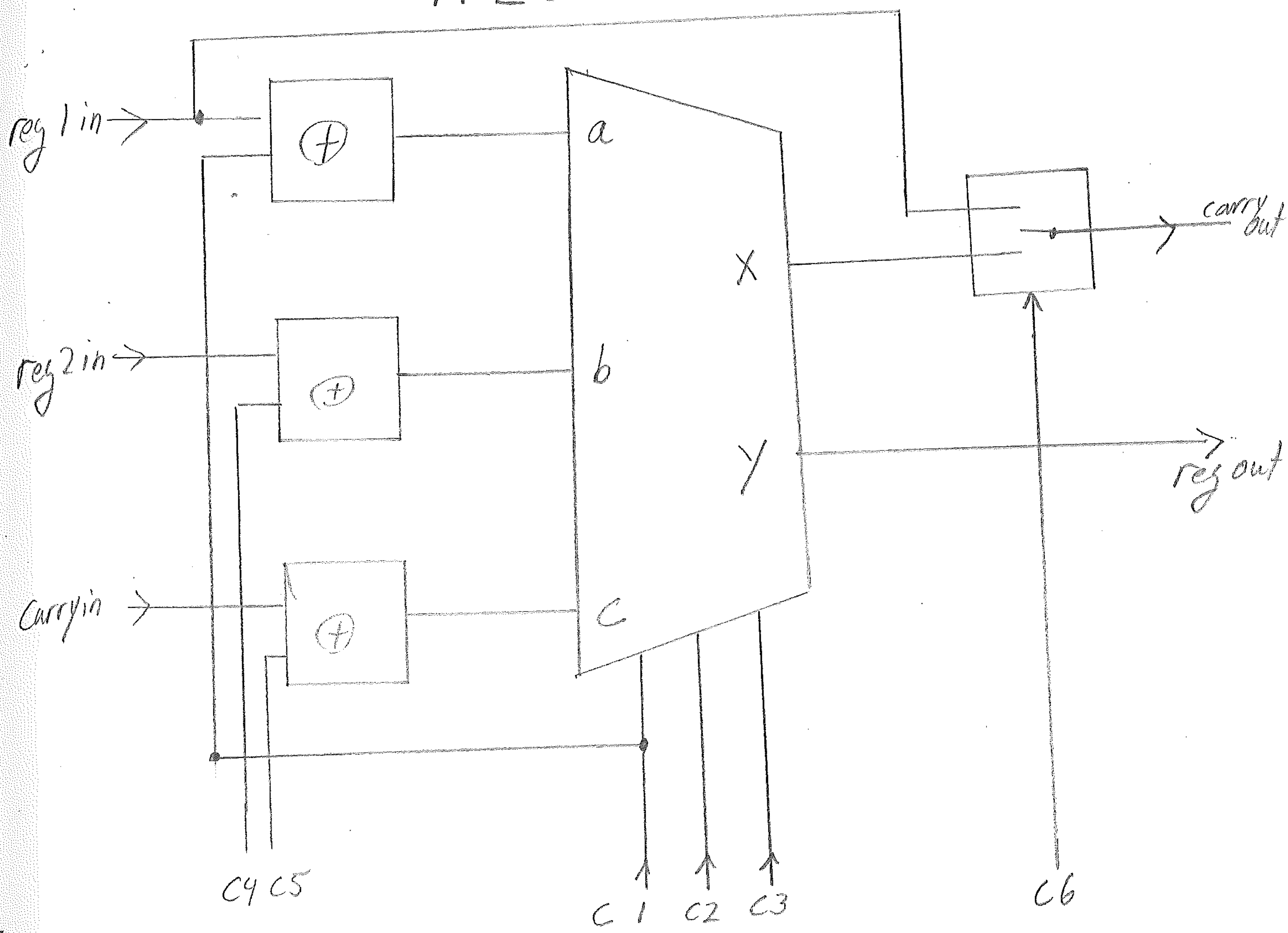
3 6 9 12 15 0

1 4 7 10 13 2 5 8 11 14

0 1 2 3 4 5 6 7 8 9

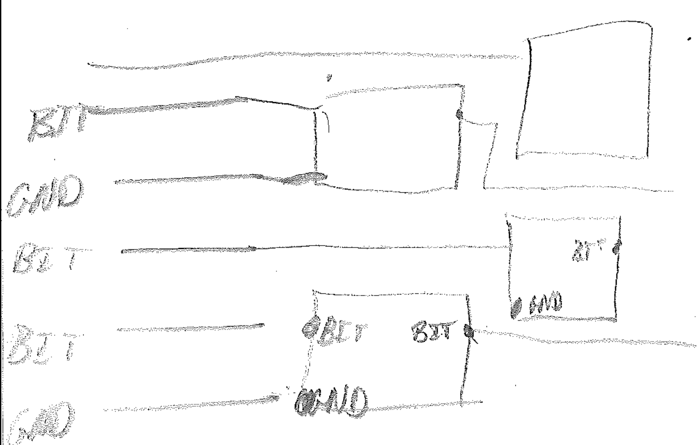
maj(ABC) = $(A\oplus B)C + AB$

ALU



instruction set

C1	C2	C3	X	Y
X	0	0	$a \oplus B \oplus c$	maj(a b c)
X	0	1	$a \wedge b \wedge c$	$a \wedge b \wedge c$
X	1	0	$a \vee b \vee c$	$a \vee b \vee c$
0	1	1	b	$(a \oplus B) \wedge c$
1	1	1	c	b


BIT
GND
BIT
BIT
GND
GND
BIT
BIT
GND

Stix

(load "ml:minsky;stixed")

(:stix-edit)

purple = buried

x-help

x-thick - expands to real thickness

<help>

x-translate goes to DPL

x-clear

x-scale

(center-mouse)^2 = center at that point

(center-mouse)^3 = center object

^e edit part

^p instantiate a part

^x x-dim compaction

restart

(chip-init) clears buffers

(stix-edit) startup

can use ~~#~~ *.press forms yeah

BKcell
BKcellf
BKchip

Empress <#>/e fn.press

redraw/m
merge

Pressedit $lides.Pres ⇐ foo.pres . . . . . .

destination filename

Alto

Programs
Draw
redraw
empress
neptune ~ Dived

calling (Draw 8/g gives finer grid

esther's password
Frob[illegible]le

file types

.draw } redraw
.press }
» dover by empress

circle.draw has circles in it.

in draw
↑g grid for boxes
↑t changes text curser placement
<Tab> ≡ ↑L

left button = exact point
middle button : ~~point~~ on ~~near~~est object
Right button = ~~straight line~~ grid point nearest cursor

↑w ~~write~~ write in file trash.draw
↑r read in file ~~trash~~.draw
↑u undo
↑Q Quit

Pads in "schip3;pads"

iopadbuffered

poly
diff
metal

enable-input bottom (use it as input)
enable-output

ALU PLA Description

1) two cases decode instruction set and select among all of the ~~ ~~ ie 2 plas

2) or one large PLA

1) instr PLA inputs C1 C2 C3

outputs: 1 is high if $\overline{C2} \cdot \overline{C3}$

$\overline{C2}\,C3$

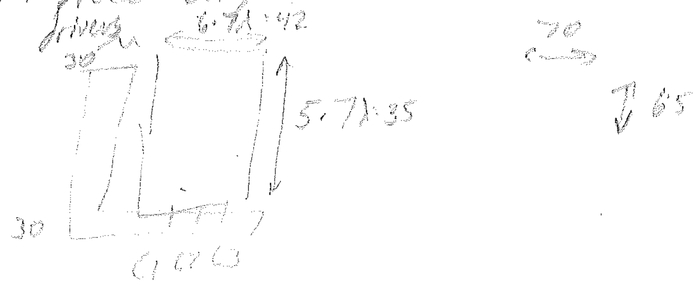
2

3 $C2\,\overline{C3}$

4 $\overline{C1}\,C2\,C3$

5 $C1\,C2\,C3$

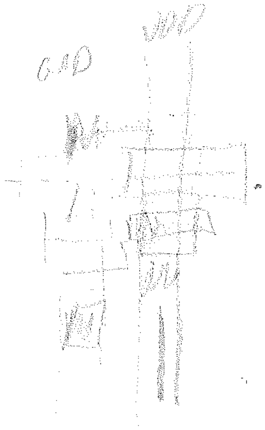
∴ need only AND Plane

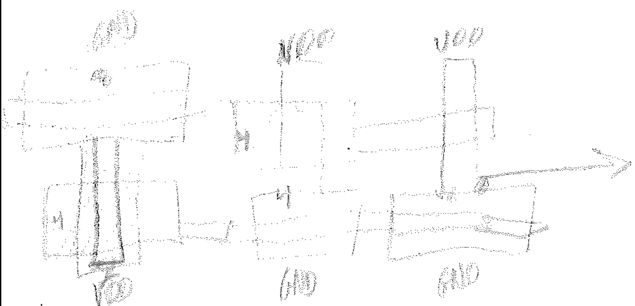


compute PLA

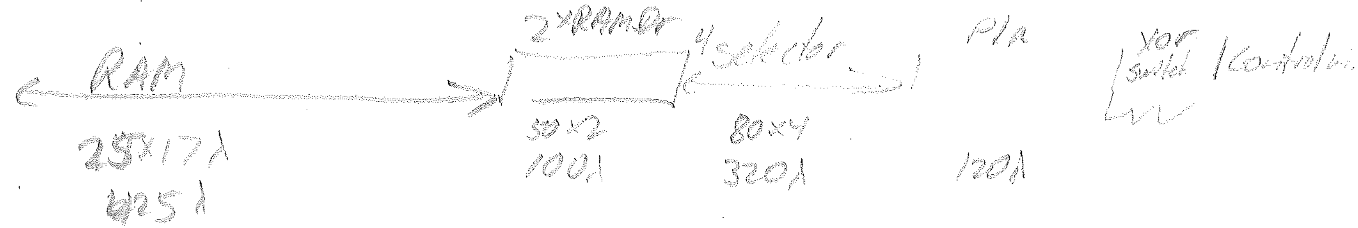
1 < (Break) >

page +2


GND
VDD

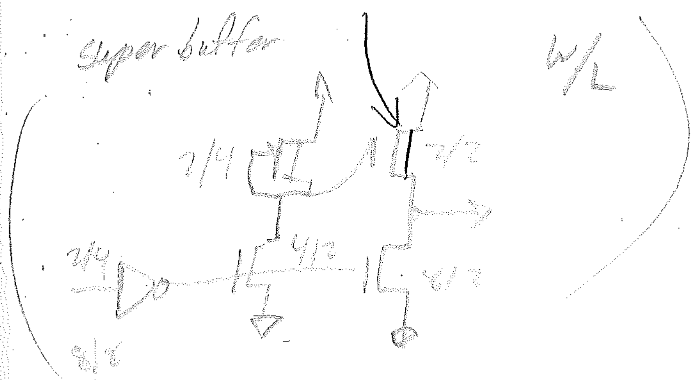

GND
VDD
VDD
GND
GND
GND

Super Buffers & BRAM Driver

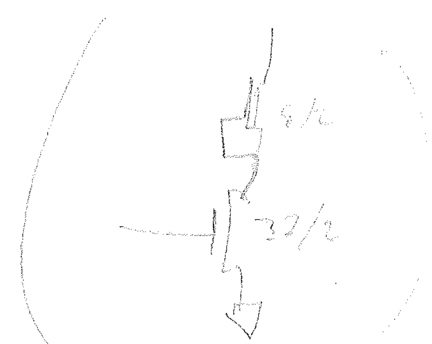


MISSING ON PROTO

96 ø



or



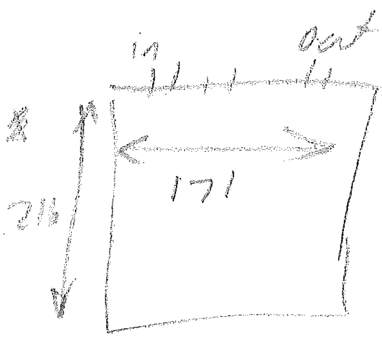
ALU PLA

$D = A \oplus B$

	min terms	out of And plane lines
$X = \overline{C2}\,\overline{C3}\, D \bar{C} + \overline{C2}\,\overline{C3}\, \bar{D} C$	2	1, 2
$+\ \overline{C2}\, C3\, abc$	1	3
$+\ C2\, \overline{C3}\, (a \vee b \vee c)$	3	456
$+\ \overline{C1}\, C2\, C3\, b$	1	7
$+\ C1\, C2\, C3\, C$	1	8

	min terms
$Y = \overline{C2}\,\overline{C3}\,(DC \vee AB)$	2
from X: $\overline{C2}\, C3$; $C2\, \overline{C3}$	
$\overline{C1}\, C2\, C3\, DC$	1
$C1\, C2\, C3\, b$	1
	12

Size:



24
17
168
24
408

↑ calculations
selector 16×5 = 80 ×4
= 120
16
ALU 24×17 = 408
= 50
= 24
RAM
RAM drivers
Flag
1000

40 | 1000

175

sender chip
ready send req
rec ready
send complete
rec ack

4sel | 5Pla | 4sel | 6ALU | 4sel | 0flg | 4sel = 27bit instruct from [illegible]

1 if "or" happens PO Box enable

4 inst POBOX to override if "or" happens

4 ϕs

3 gradual (input)

1 bias [illegible]

2 global bit out
   1 to light
   1 to rest of machine

1 data line comd bidir
   yes your message was recieved

4 bits to select processor comc

1 control in (enable to decode)

~~1 yes your message is received~~

18

Bidir comd

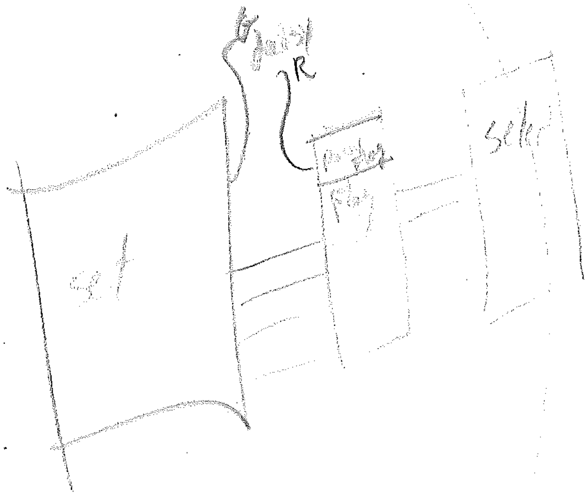
recieve: Stand 24 data
               24 addr
               1 message [illegible]

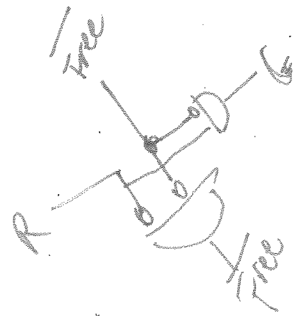
reciever enable in
   recieve 24 data
   (destination [illegible] [illegible] [illegible])

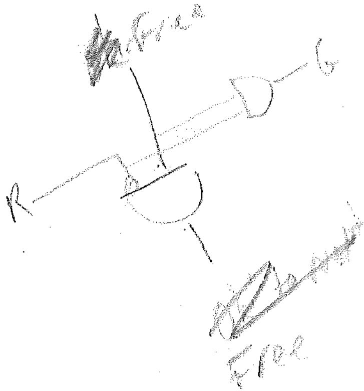
Send:
   router decides [illegible] can take one
   if no one wants it 000...0 message

Flags

Daisy Bit





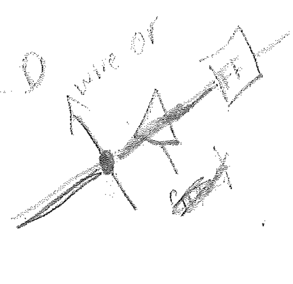


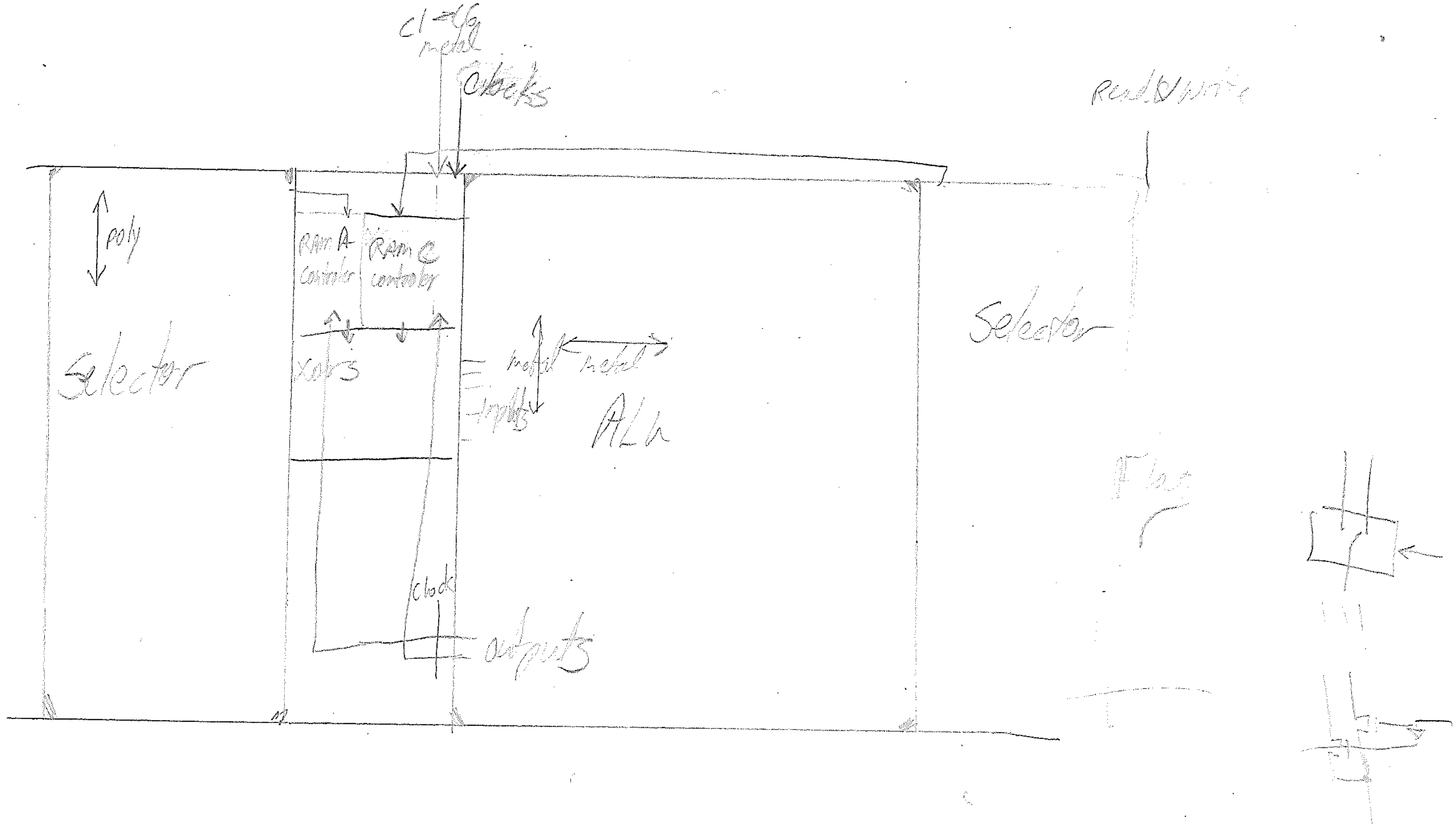
Global wire or

O Bit

Came row column decode

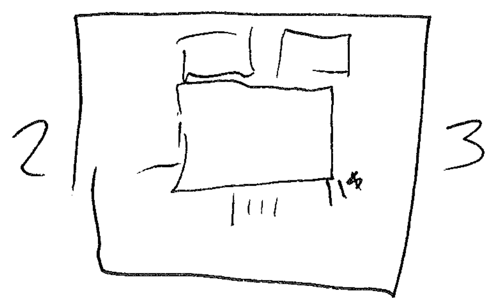
Word




c1 - c6
metal
clocks
Read&write
poly
Selector
RAM A controller
RAM C controller
xors
metal
inputs
metal
ALU
Selector
clock
outputs

"AI: RSG; Hinge load"

8



8

10

2700

RAM chip

Pads

in 4 sel A

4 sel B

4 RAM Rd + Wr

3 RAM Driver 1 per/Row

3 Ram Driver 3 per/cell

2 VDD + GND

out

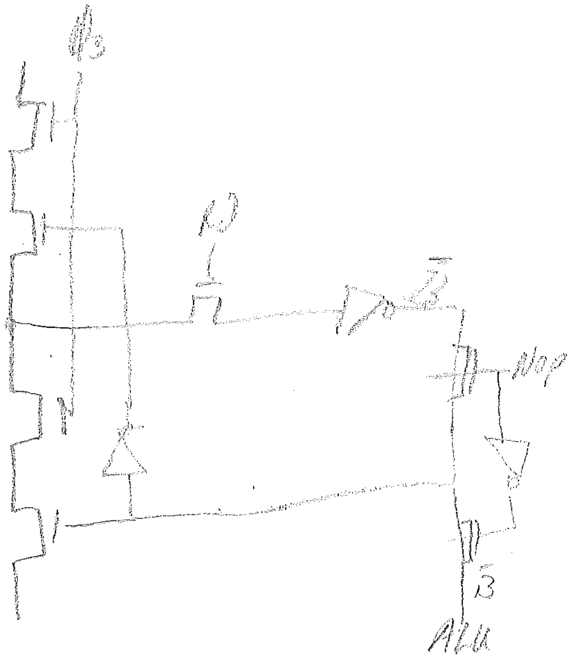
1 A standard out

1 B out

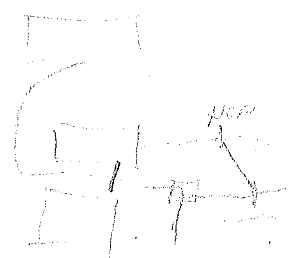
1 extra A out

---

21 Pads



ϕ_3
$\overline{B}$
Nop
$\overline{B}$
ALU



Nop

Left-connect

prechary-top vdd-top and-top

Prechary-botton vdd-botton gnd-botton

Ram-driver pair

GND-top-left read-top-left write-top-left gnd-top-right read-top-right write-top-right

GND-bottom-left

Super buffer

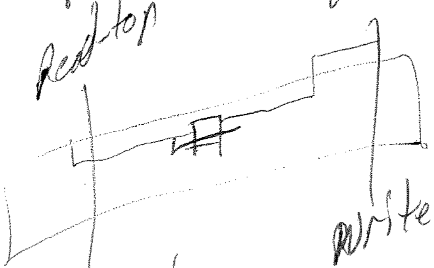
vdd-connect-top GND-connect-top

in

out

Ram-cell-6

Read-top write-top



Write-botton

Read-botton

special-(left / right)-selector

~~Flag~~

selector

in-1 . . . in-16

super-buffer GND-connect-top vdd-connect-top . . .

GND-connect-botton

Play - Rewrite-driver

Read-top GND-top-Left GND-bottom-Left vdd-top phil-top write-top VDD-top ~~VDD~~ GND-top-RIGHT

a-out (poly contact)

a-in (diff)

special-ram-with-drivers

gnd-line

GND-~~2~~-top
read-1
write-1-top
gnd-3-2
read-2

16 - 1

1 - 16

GND-line

GND-connect

Small-selector P/a

use 2 poly

in-1
in-2
in-3
in-4

VDD-connect

out-1 out-2 - - out-16

Pads.

4 B sel-in

4 or 5 with Ram Stuff

4 A sel-in

phil-in
a-out

Precharge-in
Bout

ALU-in
NOP

read-in
write-in

8

8 in

in phi
in Bit-in
out

3

2

in
out

~~wr clock~~

4 in

RAM

in wr

in rd

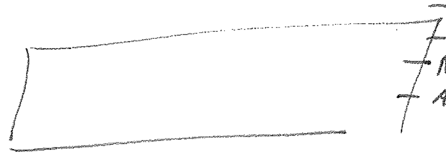
NOP

6

Pads

B-select~~or~~-4 B-select-~~3~~ ... B-select-~~4~~ A-select-4 . . . A-select-~~1~~

- wr
- Rd
- Phi1
- NOP
- ALU-in



Precharge
B-out

write
Read
Phi1 -
a-out
~~bit-out~~

RAM-read-1
R~~am~~-write-1 Ram-read 2 Ram-write ~~2~~ Bit-in NOP

VDD – BOT. –23, –47

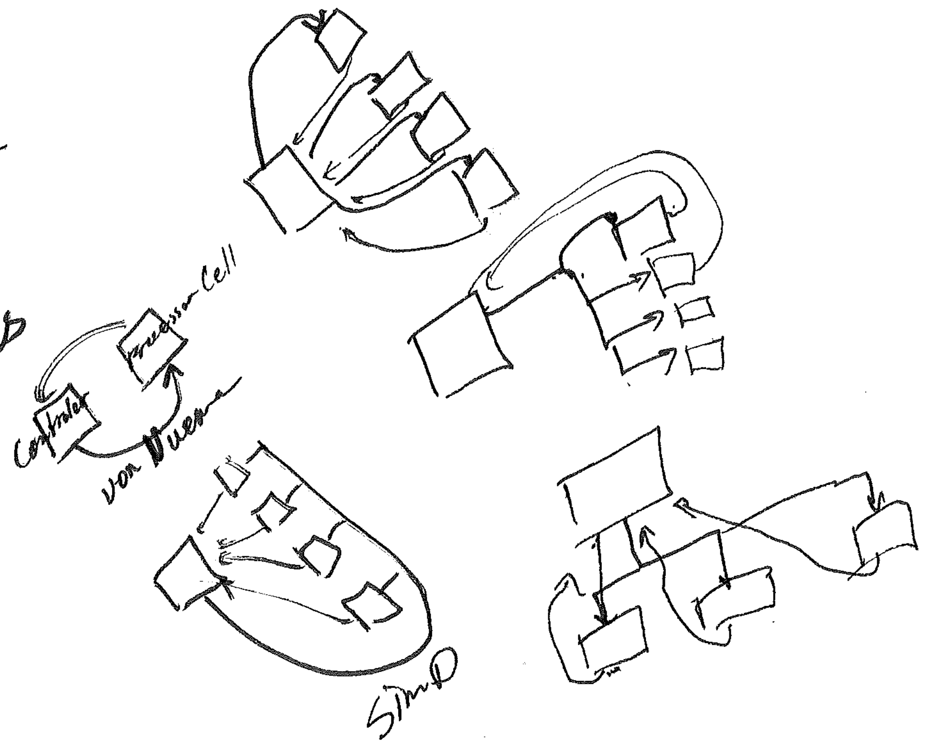
VDD – TOP –23, 13

Need

~~conventional~~

~~SIMD~~

machines



controler

Program
+
~~test~~
execute

Von hardw

SIMD

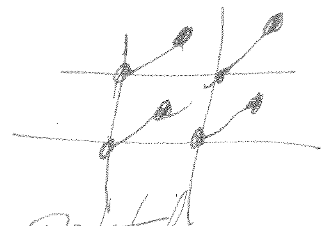
Thesis:

Alto diagrams

✓ bkalu.draw
✓ bkcell.
✓ bkcellf.
✓ bkchip.
✓ bkcm
bkcmf
bkglobal
✓ bkram
bkramdr.draw + gil

dalu ✓
dcell ✓
dcellf ✓
dchip ✓
dcm ✓

draw ✓

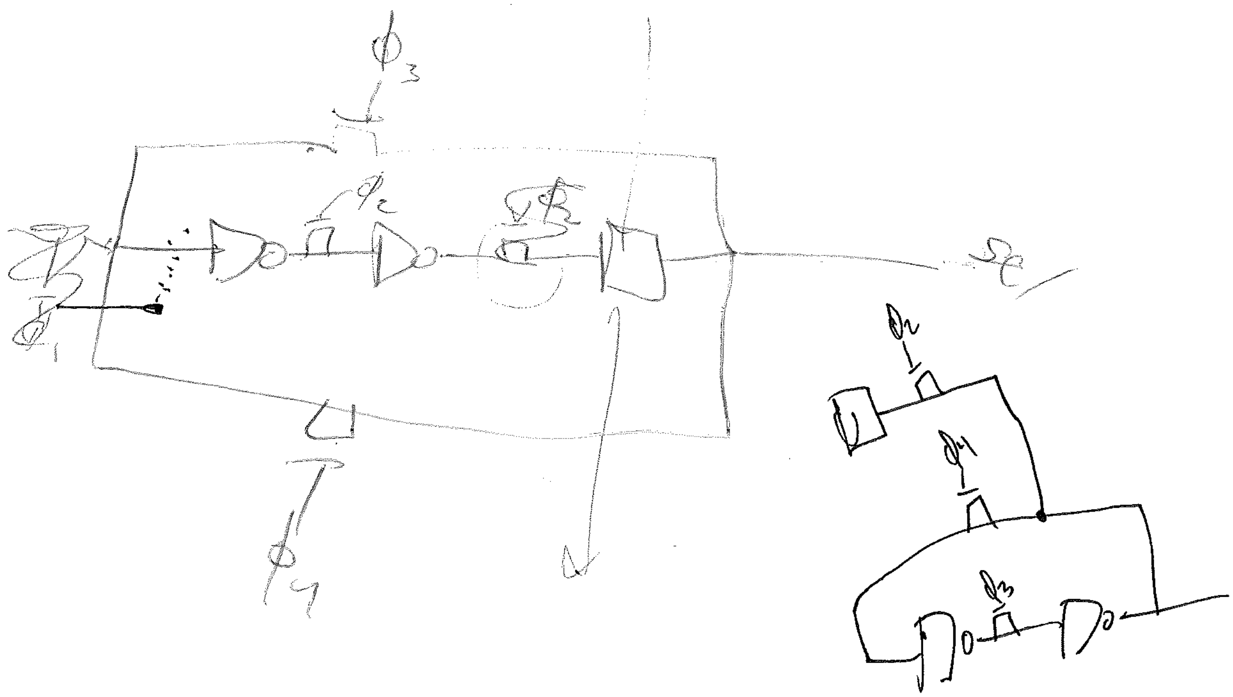


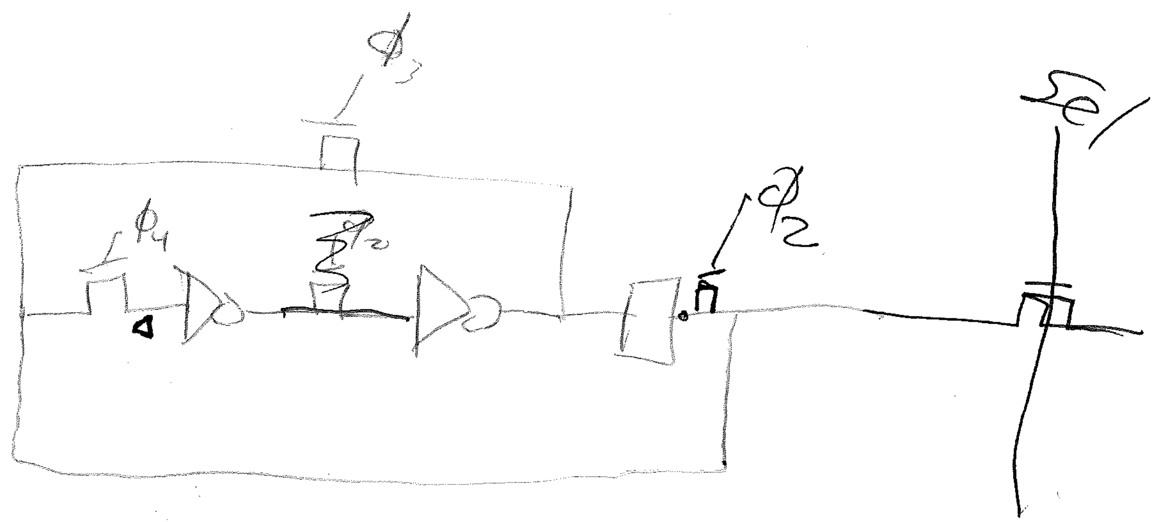
~~global~~

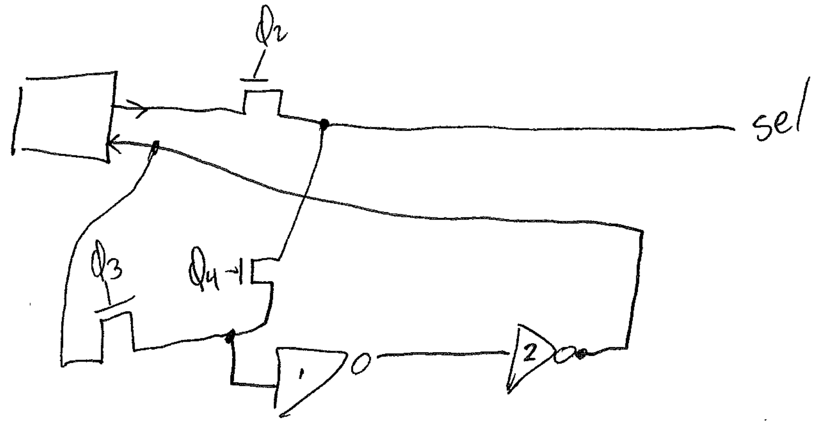
bkconn.draw

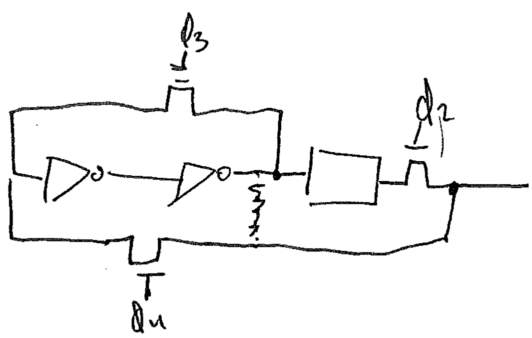
connection machine = dcm

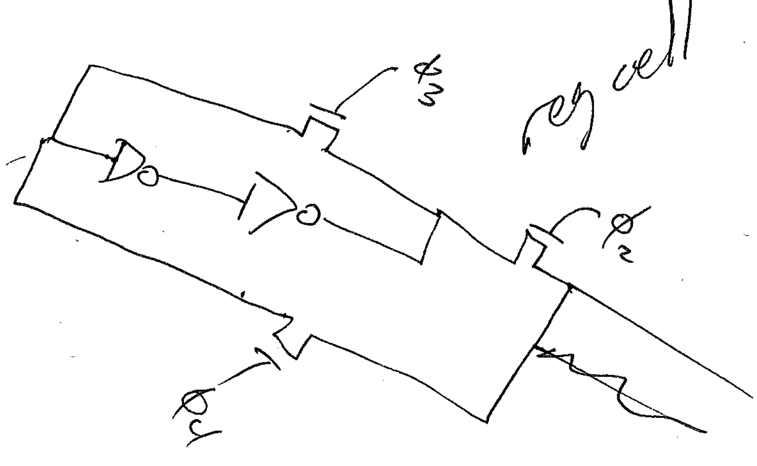
Bk Von.draw

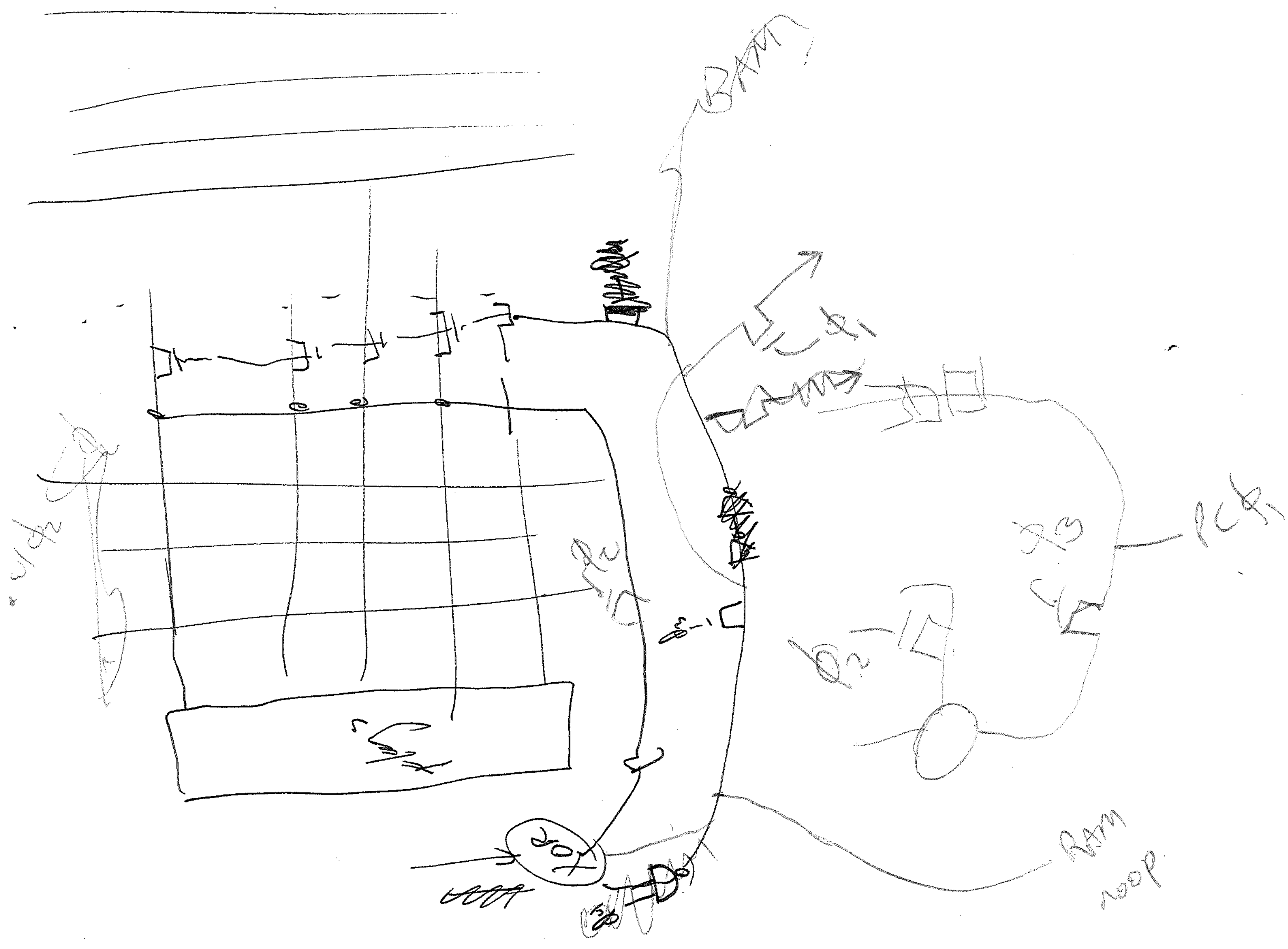

ϕ_3
ϕ_2
ϕ_2
ϕ_1
ϕ_4
sel
ϕ_2
ϕ_1
ϕ_3


ϕ_3
ϕ_4
ϕ_2
ϕ_2
sel


Φ2
sel
Φ3
Φ4
1
2


Φ3
Φ2
Φ4


Φ3
reg cell
Φ2
Φ2


RAM
PC φ1
RAM
loop
XOR
flags

in	in	in	bi	in	bi	in	in	in	in	in	in	in	in	in	in
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Free	GND	com C V,H,H	Global	VDD	com D	Φ_4	Φ_2	Φ_3	GND	inhibit toggle	VDD	GND	Φ_3	Φ_2	Φ_1

out	bi	out
↓	↓	↓
Free	inhibit, inhibit	ram-inhibit, ram-inhibit

bi
↓
Free intermediate

detect sel
4 in
1 out, 1 out

r/w sel
4 in
1 bi, 1 bi

36 pads

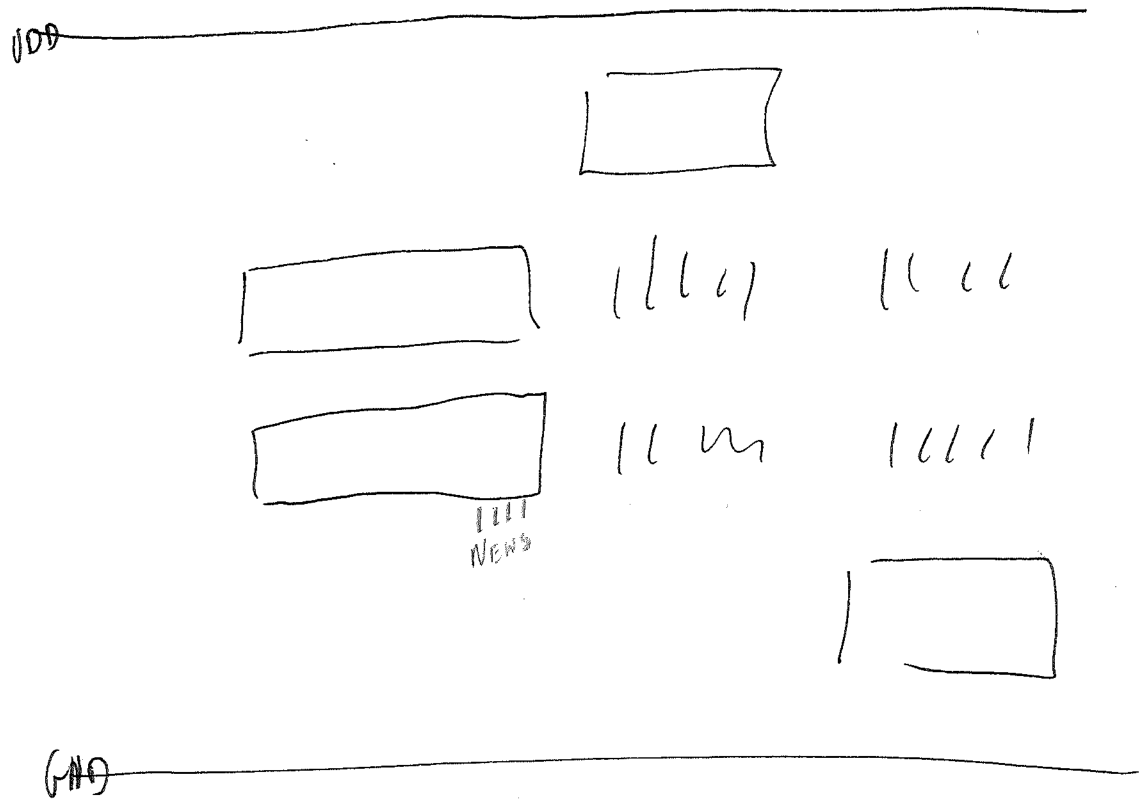
10

VDD

conc V in
conc h top in
Q_4 Q_2 Q_3
in
inhibit toggle
in
~~inhibit~~
~~out~~

sel in

GND



conD bi
conD gate
free in

free int
bi
free int gate
in

6

free
out

Global out
Global gate

inhibit bi
inhibit gate in

inhibit ram
out
defeat out

6

inhibit bi
inhibit gate
in

GND

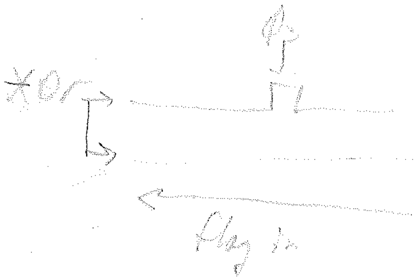
conc h bottom
in
Q_3 Q_2 Q_4
in
inhibit ram
out
defeat out

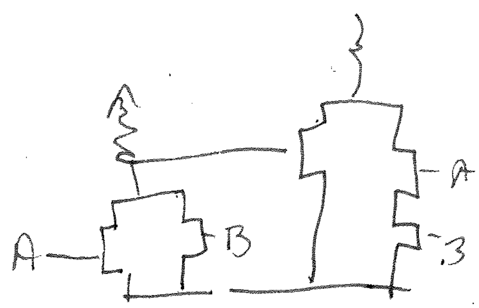
sel in

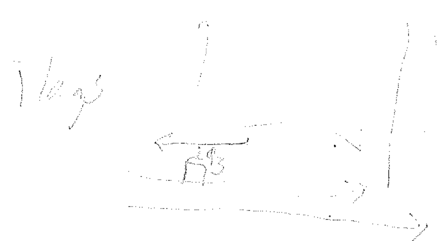
VDD

10

LAYOUT



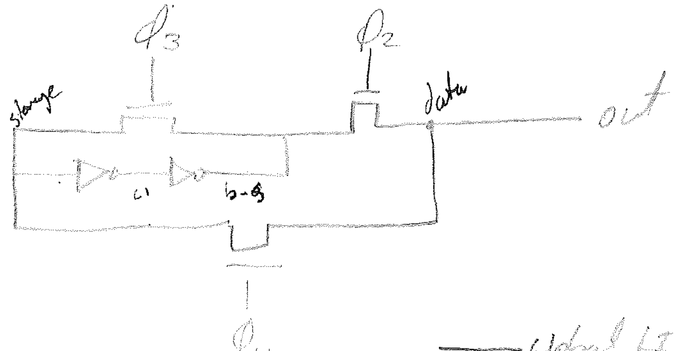




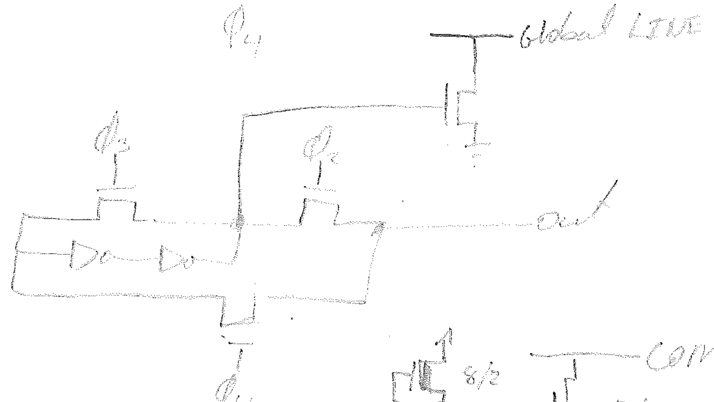
Flags 1/7/83

after phi = 2 → 0
after phi = 2 → 1

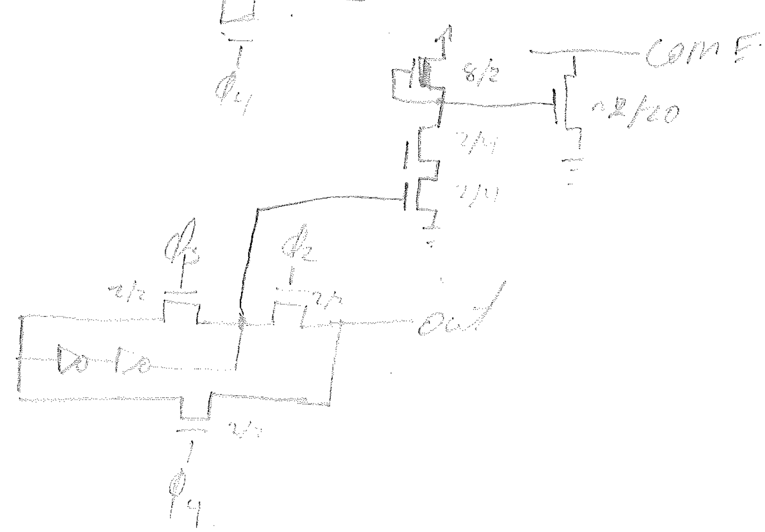
VANILLA FLAG



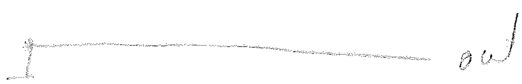
GLOBAL



COME



ZERO



Flag → Alu (out)

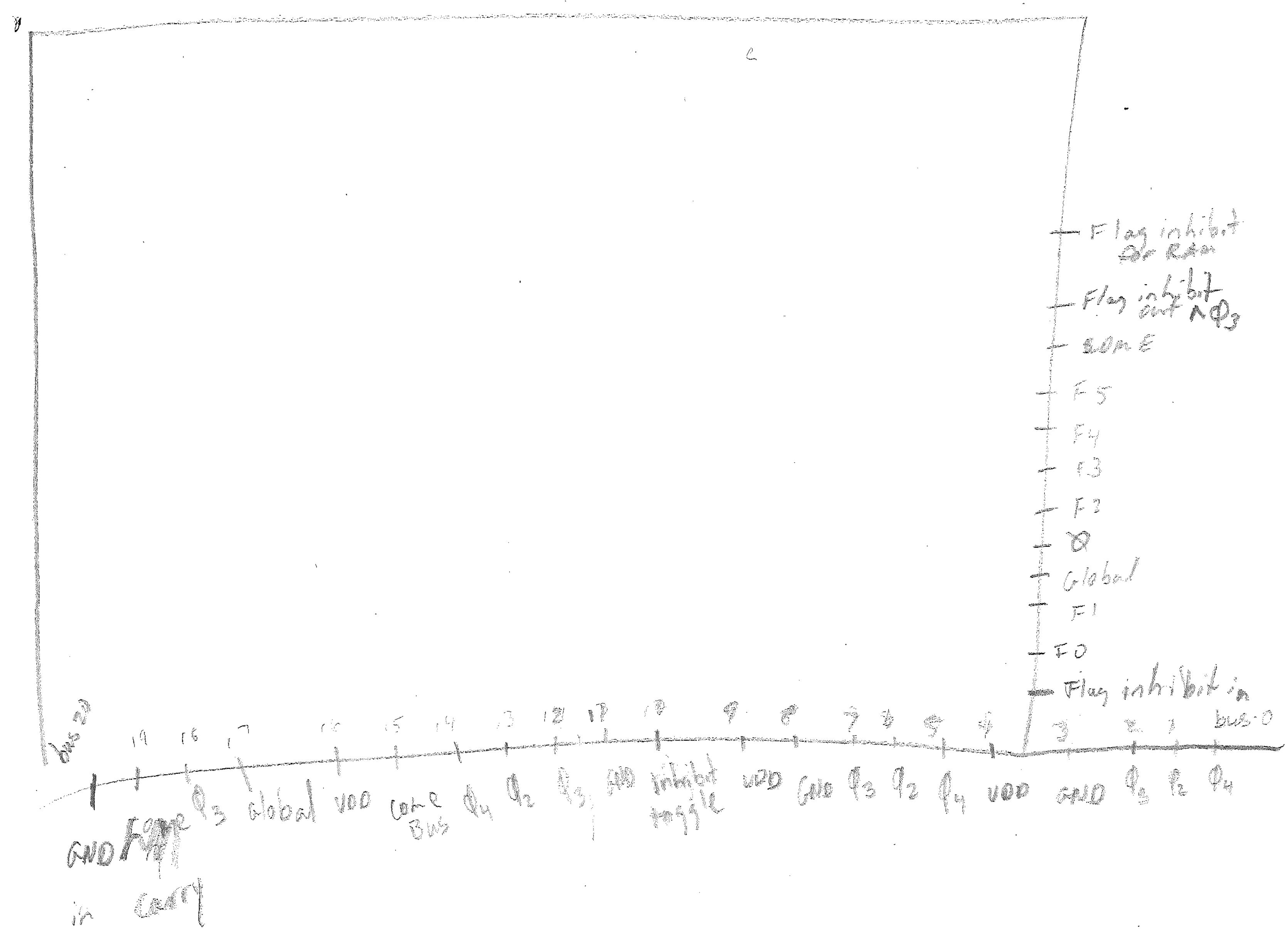
8 in ~~Am~~ sel
inh - 0 to 3   sel - 0 to 3

Flag in-out — in
~~Flag~~ bi.
Come - out
— out
Daisy
F6 — bi
F5 — bi
F4 — bi
2
F3 — bi
Comin
F2 — bi
N — out
Q — out
Global — bi
$\bar{E}$ — out
F1 — bi
F0 — bi
W — out
S — out

6 pads. inp
5
5

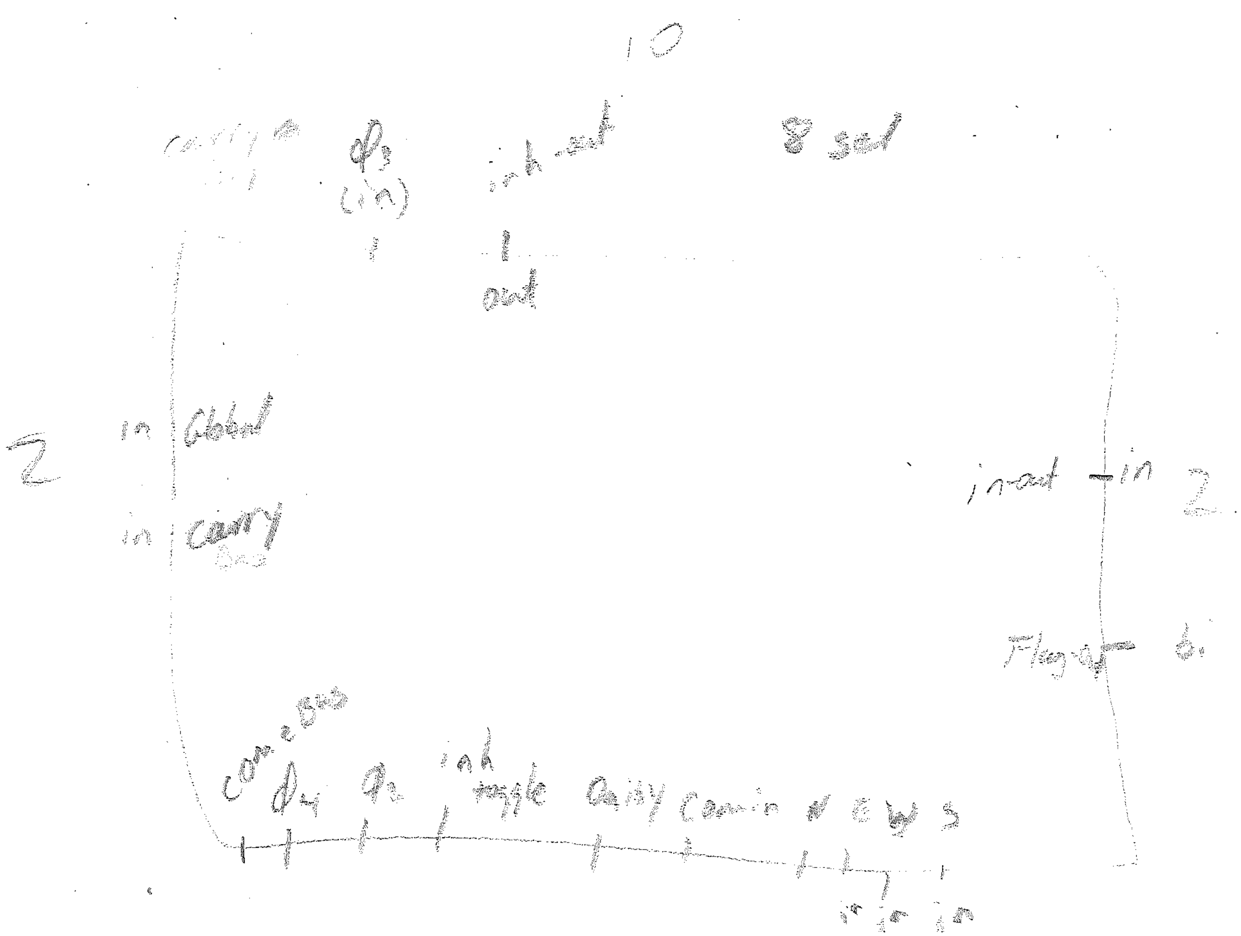
GND in
Carry in
ϕ_3 in
Global in
Come Bus in
ϕ_4 in
ϕ_2 in
inhibit toggle in
Daisy in NEWS in in in in in
inhibit
VDD in

N E BUS
1 1 1 1



Flag inhibit Rom
Flag inhibit out ∧Φ_3
Rom E
F5
F4
F3
F2
Φ
Global
F1
F0
Flag inhibit in
bus 20
bus 0
GND
Φ_3
Global
VDD
Come Bus
Φ_4
Φ_2
Φ_3
GND
inhibit toggle
VDD
GND
Φ_3
Φ_2
Φ_4
VDD
GND
Φ_3
Φ_2
Φ_4
in
carry

24 pads


10
carry
ϕ_3 (in)
inh out
8 sel
out
in Global
in carry
inout in
Flag
conn
ϕ_4 ϕ_2
inh toggle
Parity
Conn in
N E W S
in in in

Tinhead out

8 sel
in

Onyeal In
Flag b.

Carry
in

Q3
in

Global
in

Come Bet
in

Q4
in

Q2
in

ink
Toggle
in

Daisy
in

Comin
in

N
in

in

in

in

selector.lisp

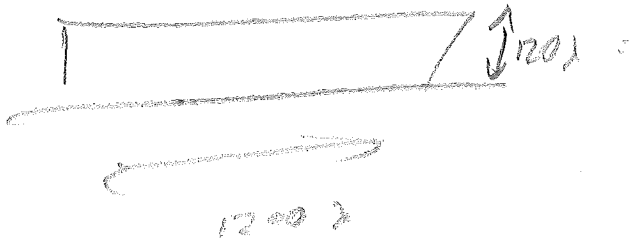
Flag-section.lisp

<Koller.cm>Chip.lisp

Chip-pads.lisp

load.lisp

<.cm>Selpk



120λ
1200λ

R-C constants in nMOS

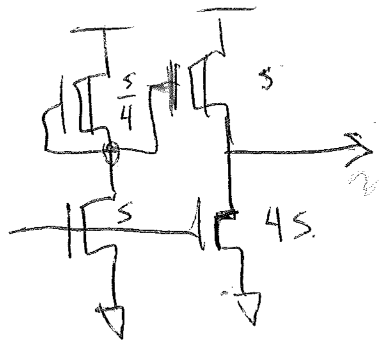
	R	C
poly	$20\Omega/\square$	6.9×10^{-3} pF/cm² ≐ 6.9×10^{-3} pF/μm²
diff field		7×10^{-5} pF/μ²
diff sidewall		20×10^{-5} pF/μ
metal		2.5×10^{-5} pF/μ²
contact		150×10^{-5} pF/contact

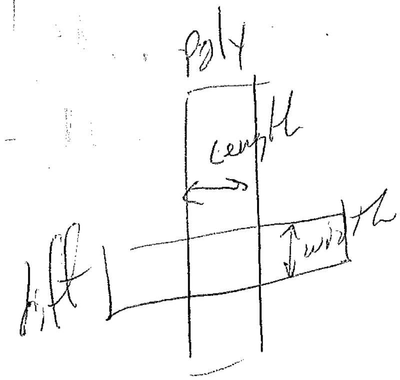
Phi 3 in-out inh-0 - inh-3 sel 0 - sel-3

in ≠ out
Chg - out

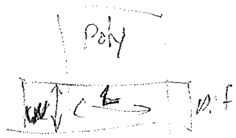
global
carry

Clone-bus phi-4 phi-2 inh-toggle daisy c-in n-in e-in w-in s-in

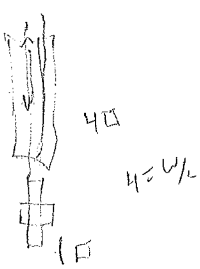




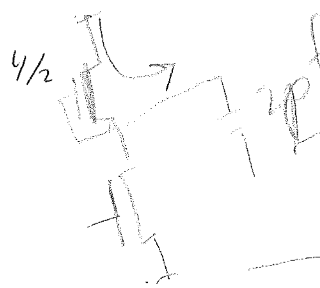
$S \triangleq \frac{W}{L}$ $\frac{L}{W}$



$4/2 \,/\, 2/4 = 4$

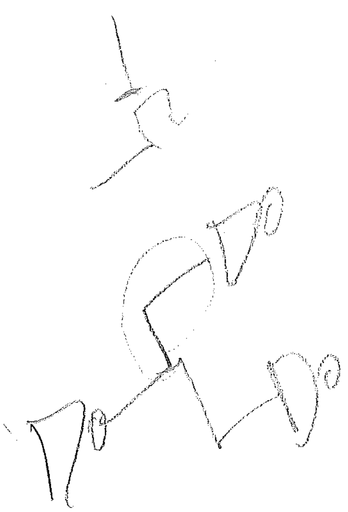


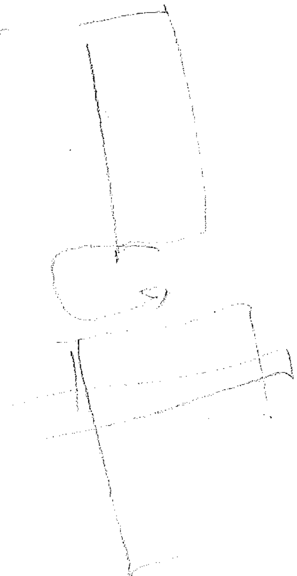
$2/4 \,/\, 4/2$

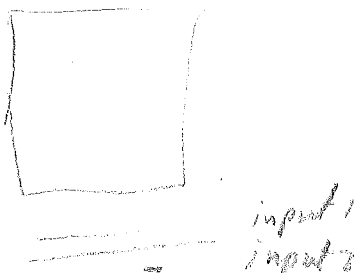


$2/4 \,/\, 4/2$

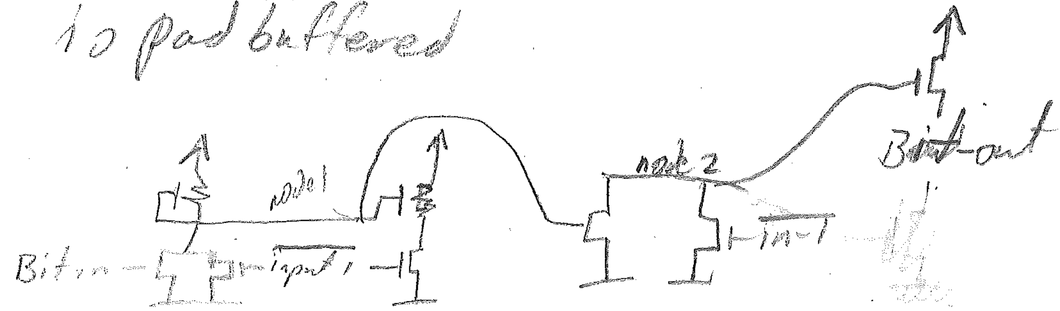
4

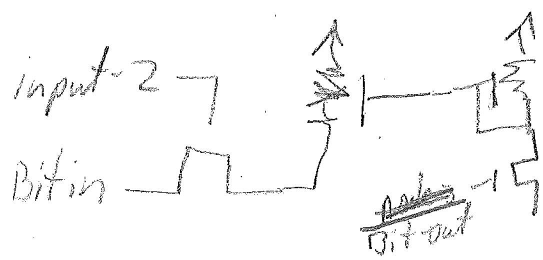






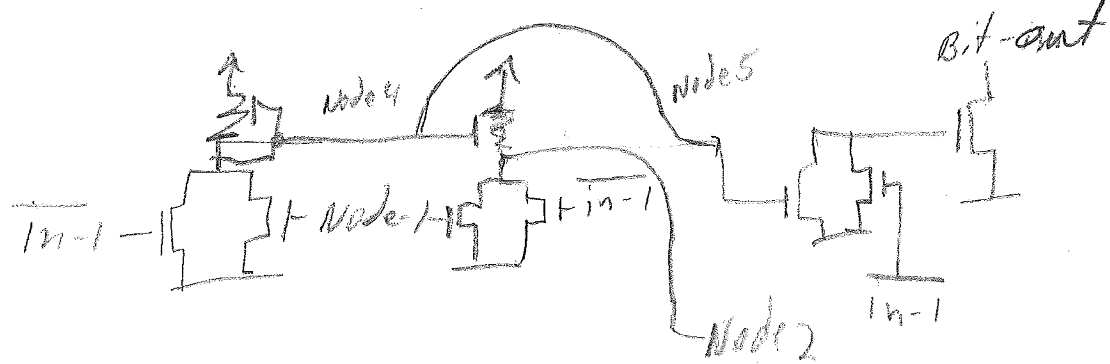
IO pad buffered





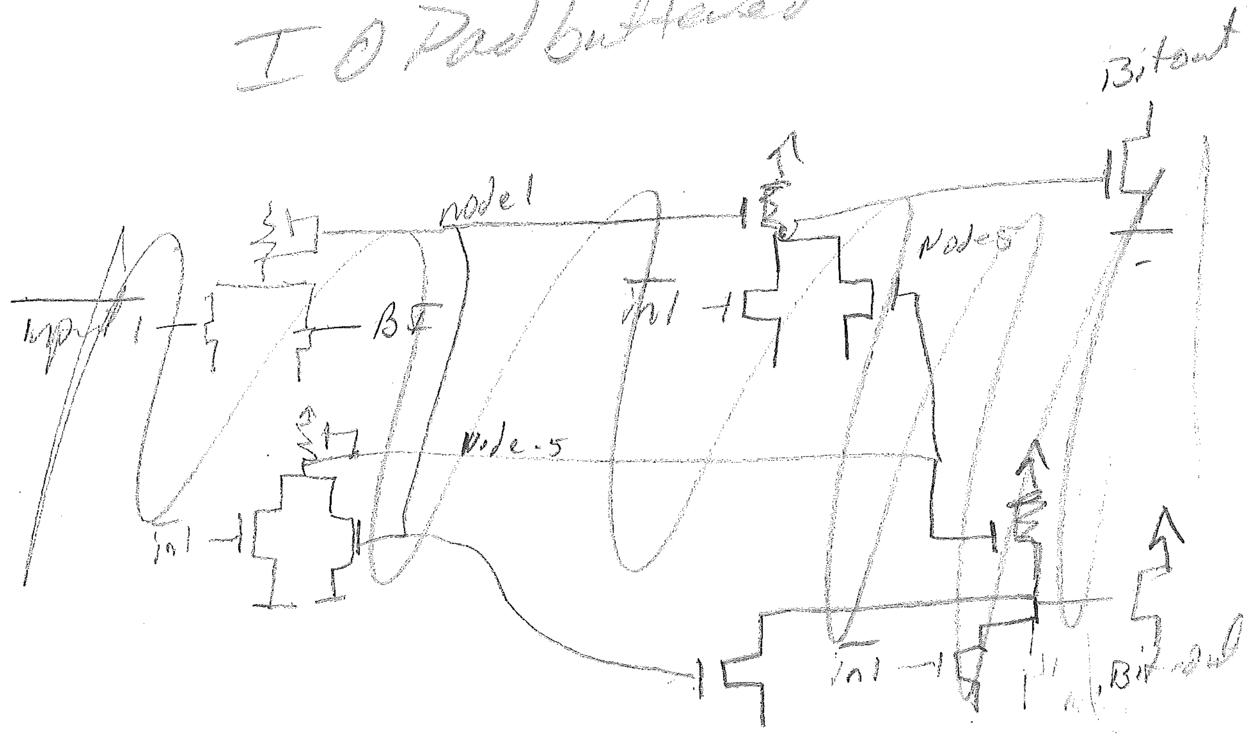
∴ input 2 = input pad enable

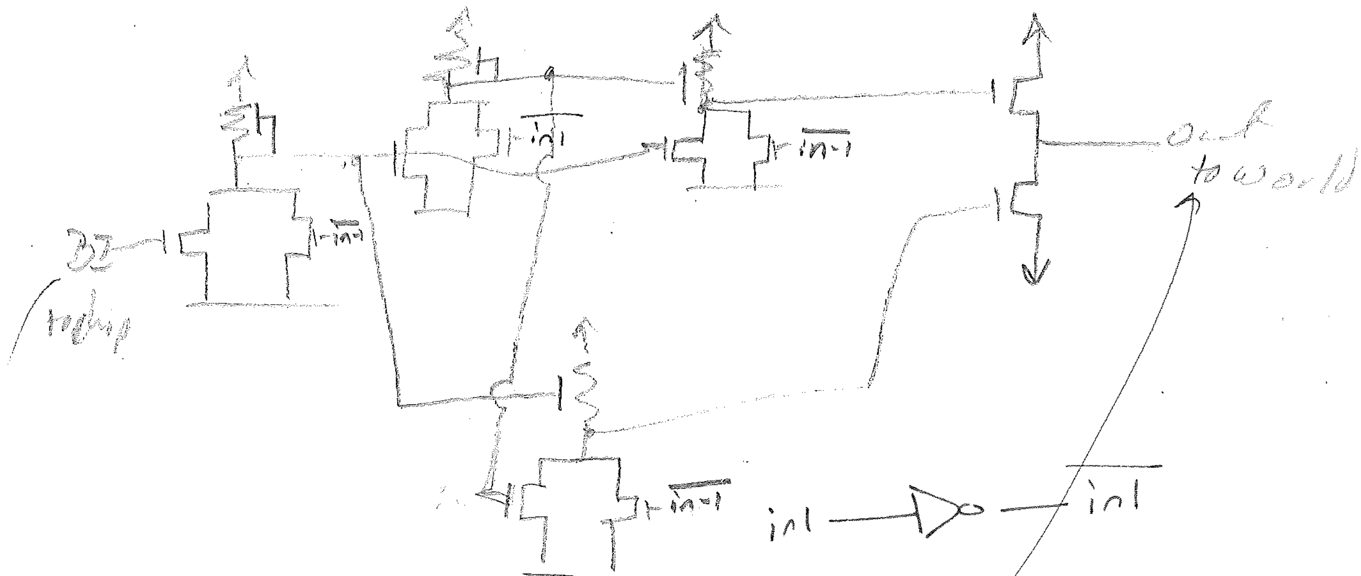
H = input pad

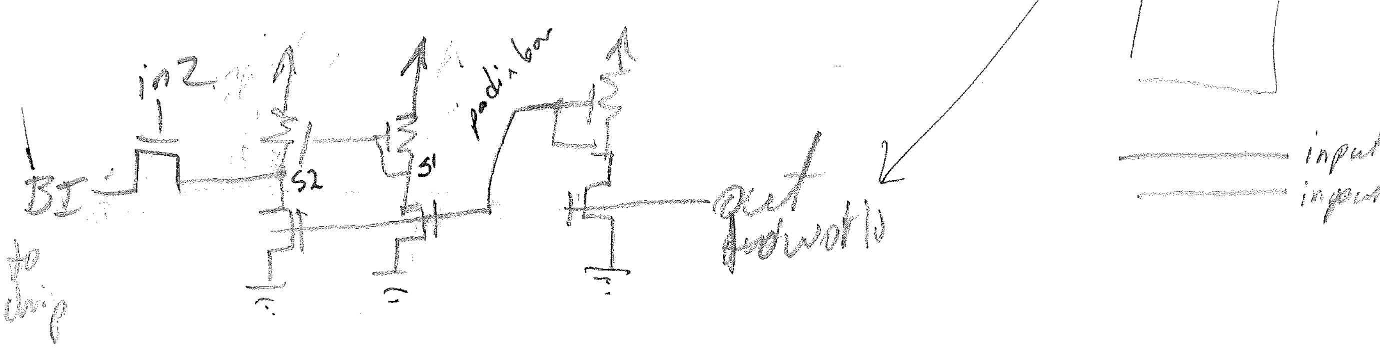


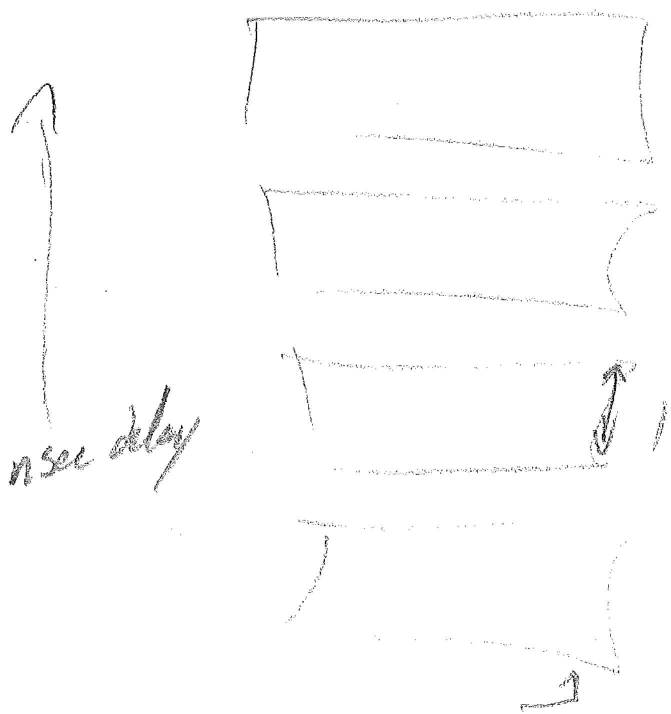
input 1 = H ⇒ output pad enable

I O Pad buffered









12 nsec delay

12.5kΩ ~~15pF~~ 3pF

4 [illegible]

ΩF = 50Ω

one
proc

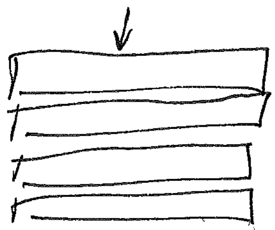
RAM Rd+write poly lines

if 4 cells high

16
5
30

375
4
1500

32 for ~~rd~~, 80 for write



$\updownarrow$ 125λ poly, 80λ gate

250 □/poly = 12500Ω
= 12.5 KΩ

at 1.5 μ/λ

$2\lambda \times 125\lambda$ poly = ~~190~~ 200μ · 3μ = 600 μ²

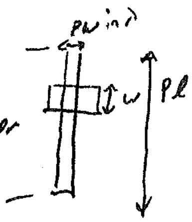
PWλ × Plλ = 2.25 PW Pl × .4×10⁻⁴ pf = 2.400 × 10⁻¹ pf

$2\lambda \times 80\lambda$ gate = 125μ × 3μ = ~~130~~ 375 μ² = 1.5 × 10⁻¹ pf

PW × 12·w = 2.25 PW · 12·w μ² · .4×10⁻⁴ pf

~~3.0×10⁻¹ pf~~

4×10⁻¹ pf



one processor

~~metal lines~~

2.25 PW Pl × .4×10⁻⁴ + 2.25 PW · 12w · .4·10⁻⁴ pf

= PWPl ×10⁻⁴ + 112 Pw · w ×10⁻⁴

Capacitance of poly lines (Pl + 112w) PW ×10⁻⁴

lengths in λ

Rd	wr
Pl = 125	Pl = 125
PW = 2	PW = 2
w = 5	w = 2
(125 + 600) 2 ×10⁻⁴	325 × 2 × 10⁻⁴
1500 ×10⁻⁴	

pf = 1.5 ×10⁻¹ read lines/processor = .7×10⁻¹ write lines/processor

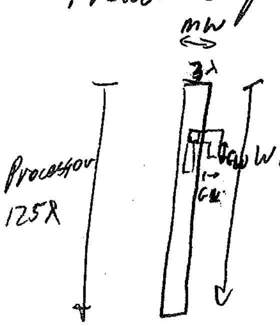
5''

12'' in²

2.5 in/in

5 cm²/in²

metal capacitance in selectors / processor



M_W width of metal $= 3\lambda$

ML length of metal $= 125\lambda$

Gw width of gate $= Gw \doteq 5$

Gl length of gate $= 2$

~~Area of poly~~

Ap area of poly $= 1\cancel{8} + GL(3+Gw)$

$\doteq$

$16 + G_L(3+Gw)$

metal line cap

$$M_W M_L \lambda = 2.25\, \cancel{M}_W M_L\, \lambda^2$$

$$= \cancel{M}\; M_W M_L \cancel{M} \times 10^{-\cancel{4}} pf$$

poly line cap

~~...~~ gate $= Gw \cdot GL \cdot 2.25\, \cancel{\times 4} \times 10^{-4}\, pf$

~~...~~ $= 10\, Gw \cdot Gl \times 10^{-4}$

rest $= Ap \cdot .4 \times 10^{-4}$

$= \cancel{8}\;\cancel{M}$

$= (7 + .4 GL(3 \times Gw)) \times 10^{-4}$

total Cap $= [\, M_W M_L\, \cancel{...} + 7 + .4 Gl(3 \times Gw)\,]\; 10^{-4}$

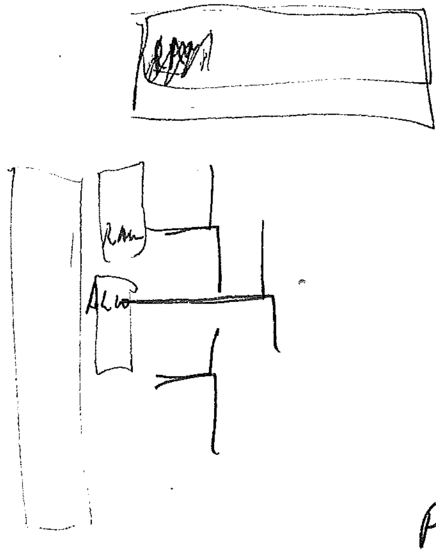
$= 375 \times 10^{-4} = .37 \times 10^{-1}\, pf$

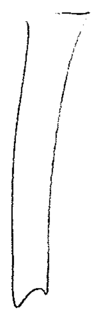
total resistance $= \cancel{...}\; \cancel{M}_L / M_W \times .03\,\Omega$

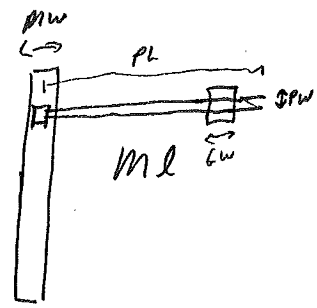
$= \cancel{1}00\,\Omega$

metal migration

1.5 mA / μ







$$Pl = 16 \cdot (MW+3)$$

$$Ml = 125\lambda$$

4000

6000 λ

ALU delays per ALU

instruction bits ~~[illegible]~~

$mw = 3$
$ml = 125$
$pl = 100$
$pw = 2$
$GW = 4$

cap metal $= mw\,ml \cdot \times 10^{-4}\,pf = 400 \times 10^{-4}$

poly $= pl\,pw \times 10^{-4}\,pf = 200 \times 10^{-4}$

gate $= 4 pw \cdot GW \times 10^{-4}\,pf = 32 \times 10^{-4}$

$600 \times 10^{-4}\,pf$ / ALU

res

metal $= ml/mw\,(.03\,\Omega) = 10\,\Omega$

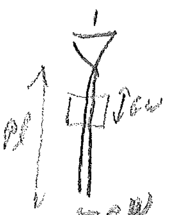
poly $= pl/pw \cdot 50\,\Omega = 2500\,\Omega$

gate = since it is at the end of the line the R doesn't matter

$= 2500\,\Omega$

decoder lines

cap



$pl = 100$
$pw = 2$
$GW = ?\ 4$ (2 for test chip)

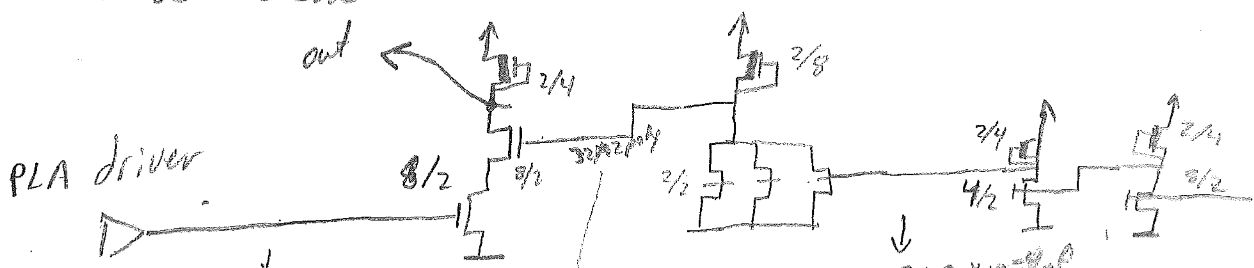
Cap: poly $pl\,pw \times 10^{-4}\,pf = 200 \times 10^{-4}\,pf$

gate
$16\,pw\,GW \times 10^{-4} = 120 \times 10^{-4}\,pf$ for $GW=4$ $60 \times 10^{-4}\,pf$ for $GW=2$

$320 \times 10^{-4}\,pf$ 260 for $GW=2$

Res
$pl/pw \cdot 50\,\Omega = 2500\,\Omega$

ALU schematic W/L for ALU test chip



$C = 600 \times 10^{-4}$ pf/ALU
$R = 2500\Omega$ /ALU
$RC = .150$ NS/ALU

$C = 64 \times 10^{-4}$ pf
$R = 800\Omega$
$RC = 5$ p

Suggest
2/4
4/2

$C = 260 \times 10^{-4}$ pF
$R = 2500$
$RC = .065$ NS

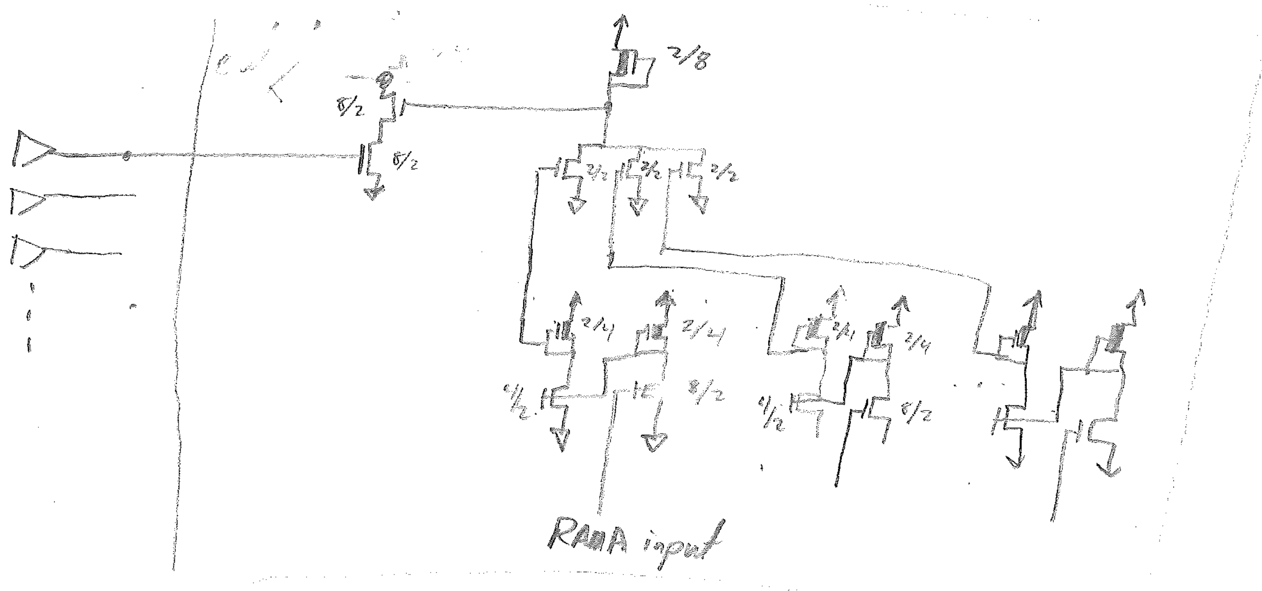
$C = 320 \times 10^{-4}$ pF
$R = 2500$
$RC = .080$ NS

$rc = \tau$

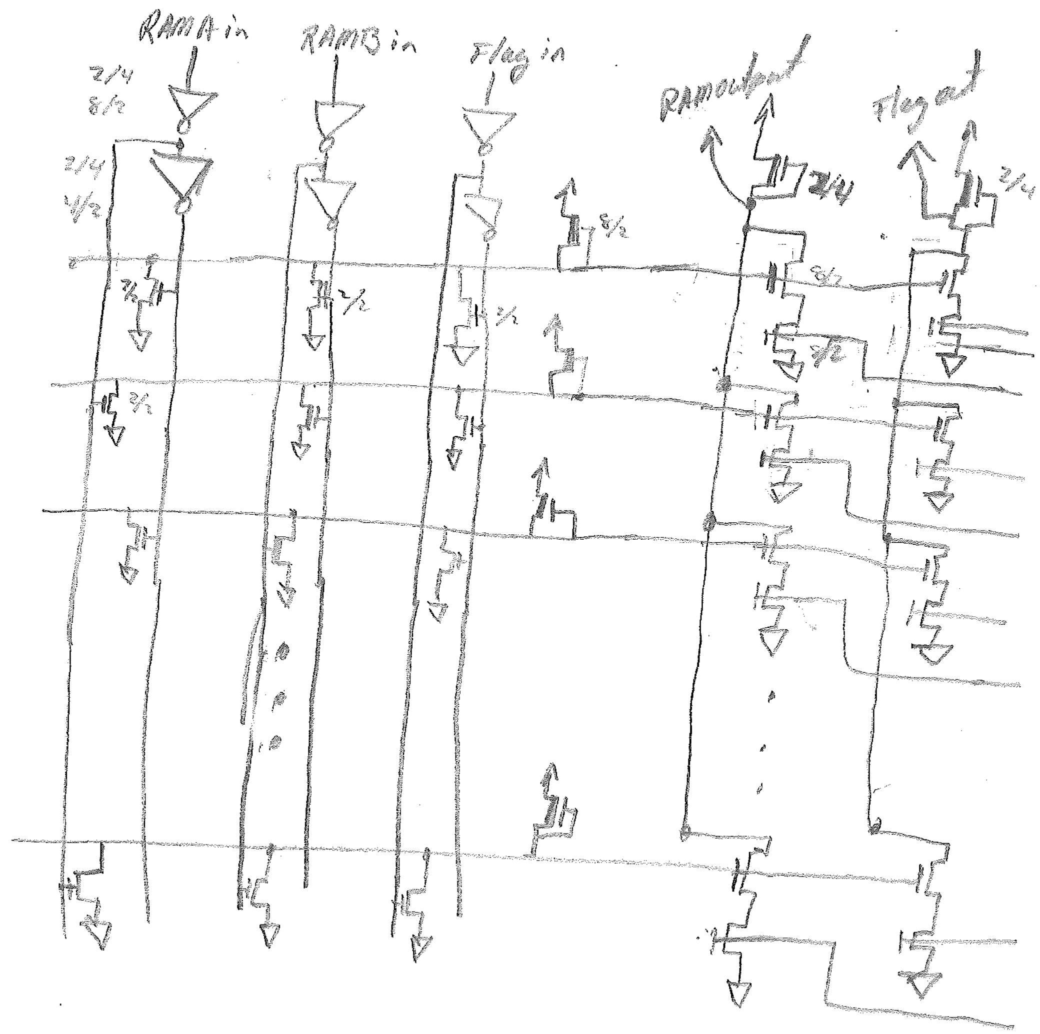
$p = 10^{-12}$
$n = 10^{-9}$
$\mu = 10^{-6}$

W/L 2/4 / 8/2, 8/2

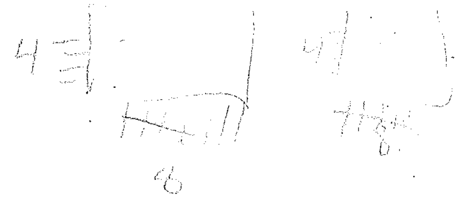
2/4/8



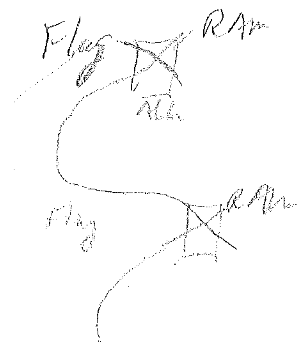
2/8
8/2
8/2
RAMA input



RAMA in
RAMB in
Flag in
RAMoutput
Flag out



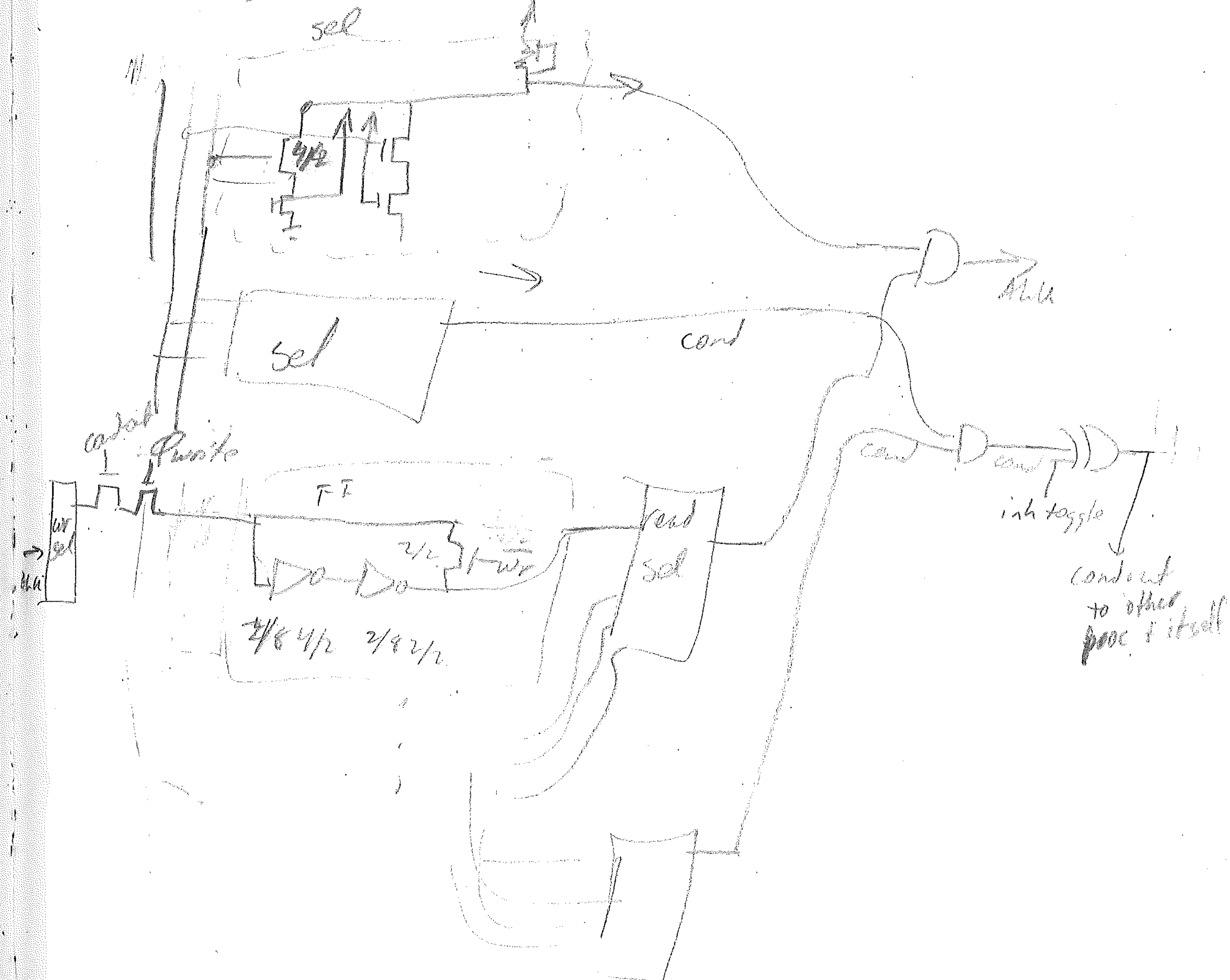
Demands on flag circuit
during Read cycle must be able
to "feed through" many ALU's
(really it's <u>before</u> write

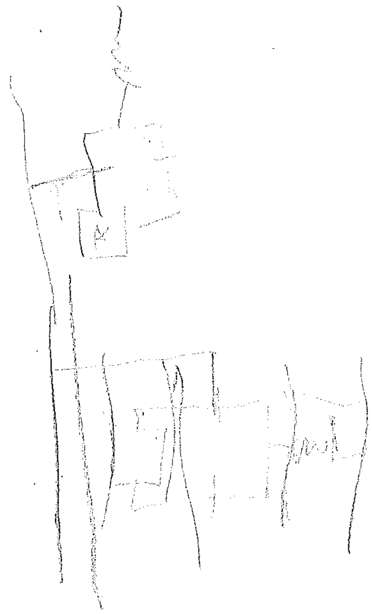


in North ⊕ RAM bit => cond out, RAM out

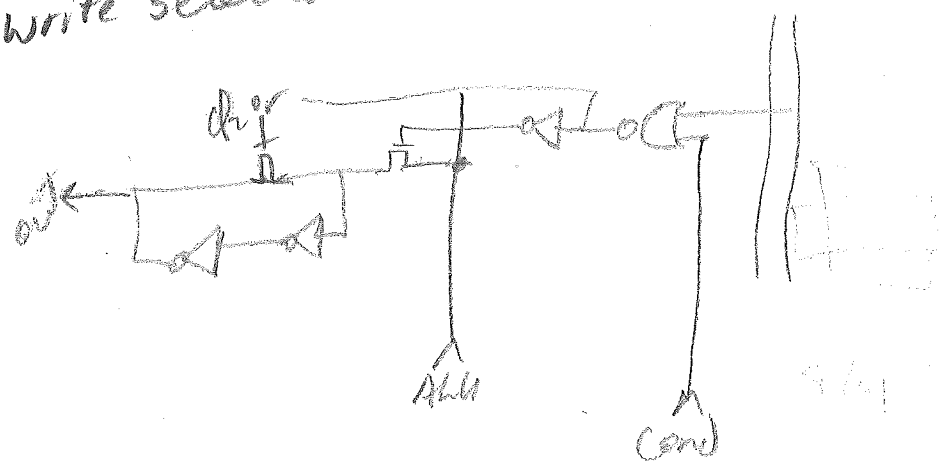
6500μx 2700μ 1.5m/λ run

Carl's Flags

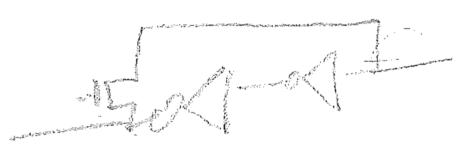




write selector

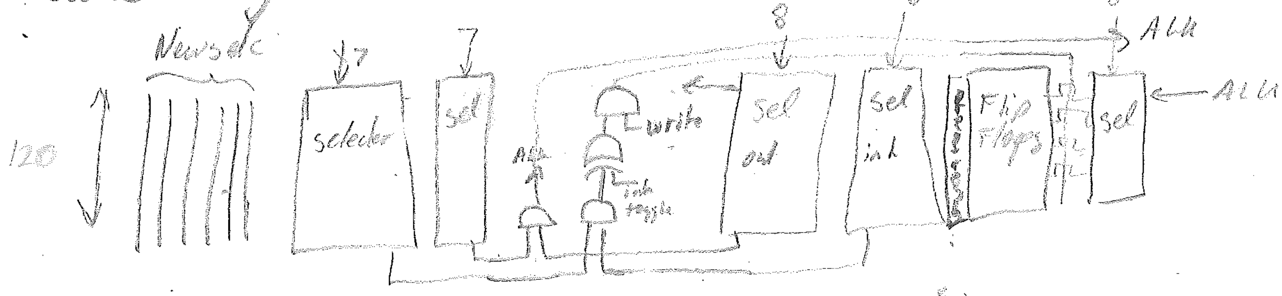


1/2 2/4



(yea)

Cards flags with decode on top



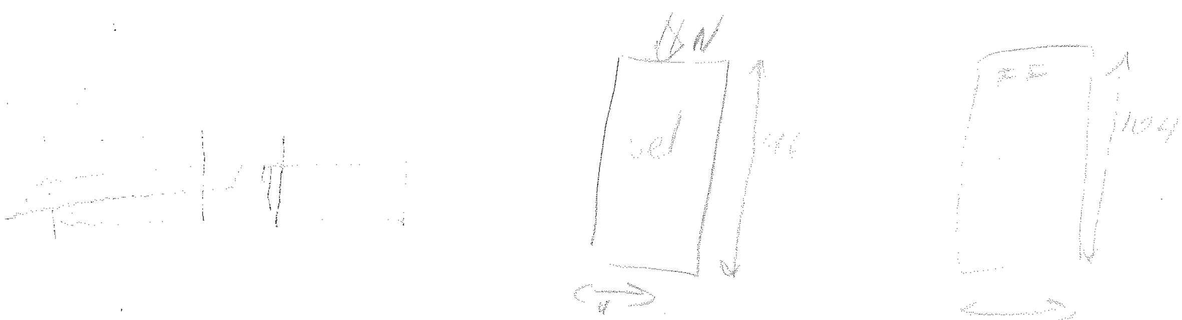
7x7 (7x7+10) x2 20? (8x7+10) x3 68 10 = 500

50 120 20 200 64

the decode on top means that if Buses are needed then one of its bits are put high, otherwise low

66
3
180

104



16*N+14

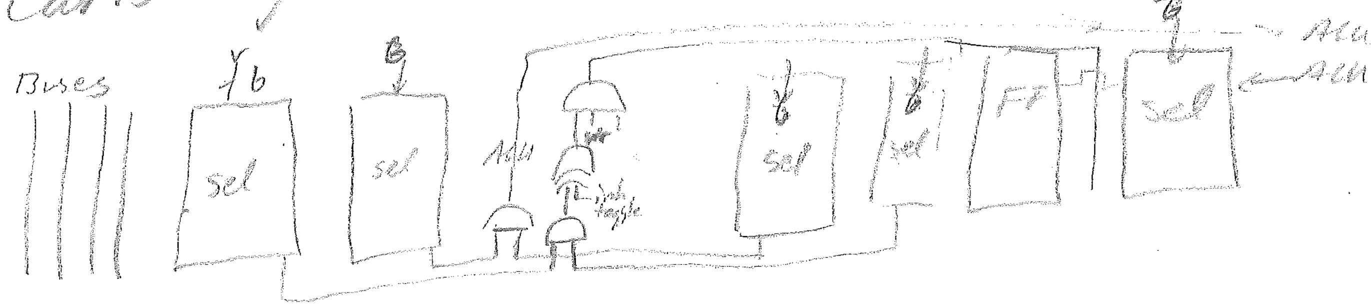
N=7 56

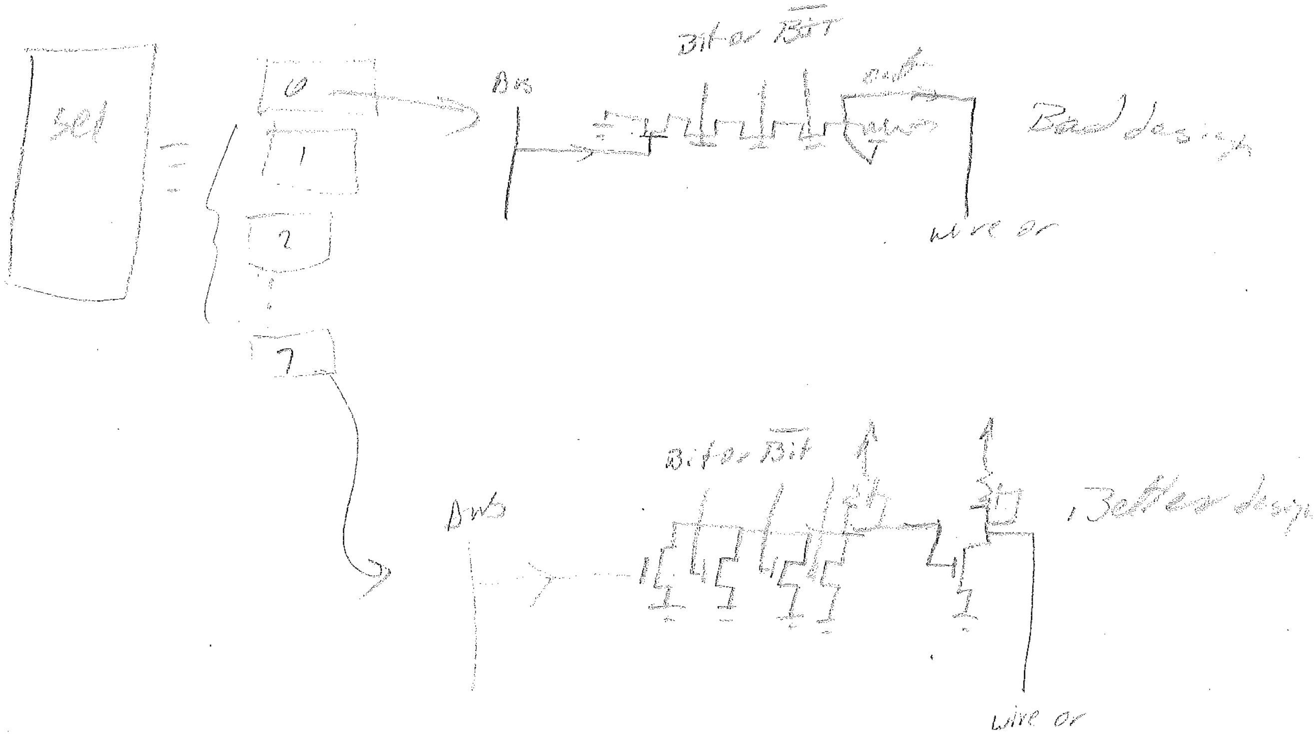
N=8 62

58

w/2 4/4 = 1

Carls flags with decoders inside (yuk)





Reasons for Debug
Test vectors
Transient errors – router
processor
Hard errors
router readout on screen case

(Bit to inhibit Global
to disable

pin : debug mode
2wide instruction

0

daisy input of
1st cell

dynamic
debuggable state bits

processors

pin enable
or
instruction

pin direction
or
instruction

self freezing
which sets a
FF that has
to be reset

static

PO Box bits

CLK
Instruction parity latch
Instruction latch
Bad wire (opt)
router readout
chip select
chip disable

want to write
~~Router broadcast bits~~

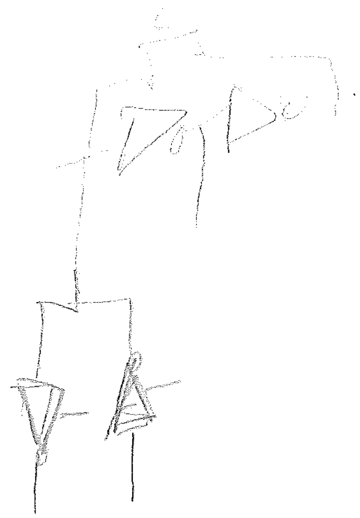
Freeze ← opt
~~Bad wire~~ ←
chip disable ← per chip
router reader ← opt
compatible

Separate gnd for pads & stuff

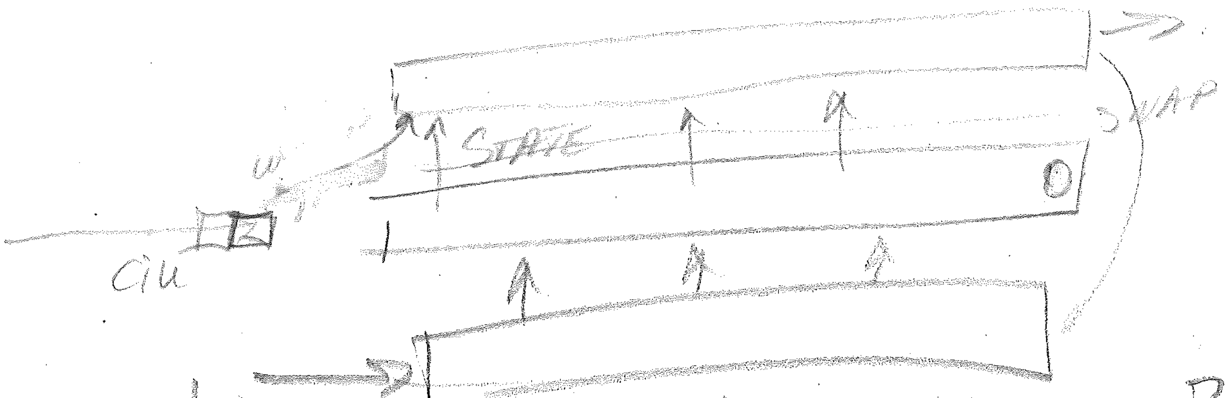
many bad pads

Same VDD

Instruction to latch into shift register to read state
Instruction to Shift
Instruction to put shift reg back



Router special mode to read state



Can these be the same?

73

Refresh cycle does not happen during Router cycle

4 News
1 RAM
1 Router
1 Instruction

On daisy input < 0 / Ring

read the ring (instruction)

make snap shot if hasn't happened

Bits in state

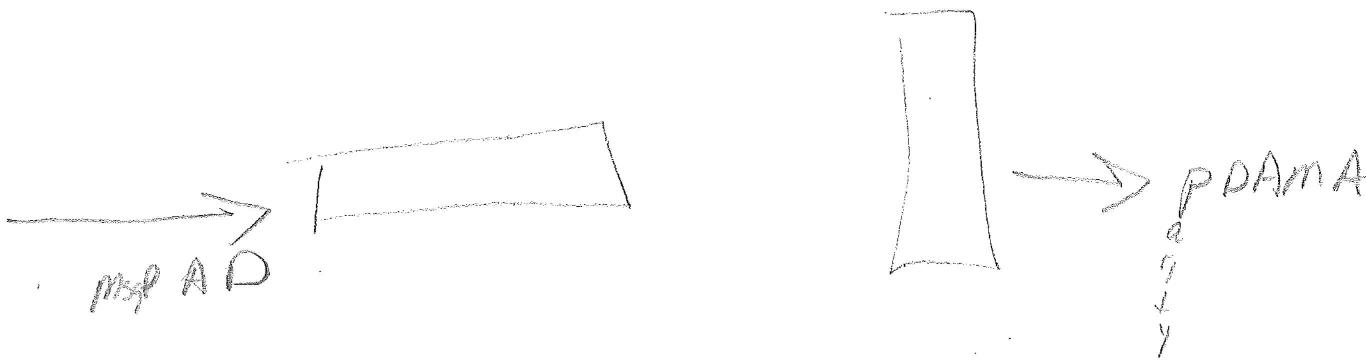
- 0 Bit
- wr Chip Select Freeze flip flop F.F — per chip
- wr chip disable FF
- wr Router "Dump your guts mode"
- wr Bad wire
- ~~back parity~~ — instruction ~~Phase~~ 1 Phase 2
- PO Bax
- CIA Flip Flops
- router misc (message, msg out, msg win
- CIA (eventually
- misc (global? News?)

Instructions

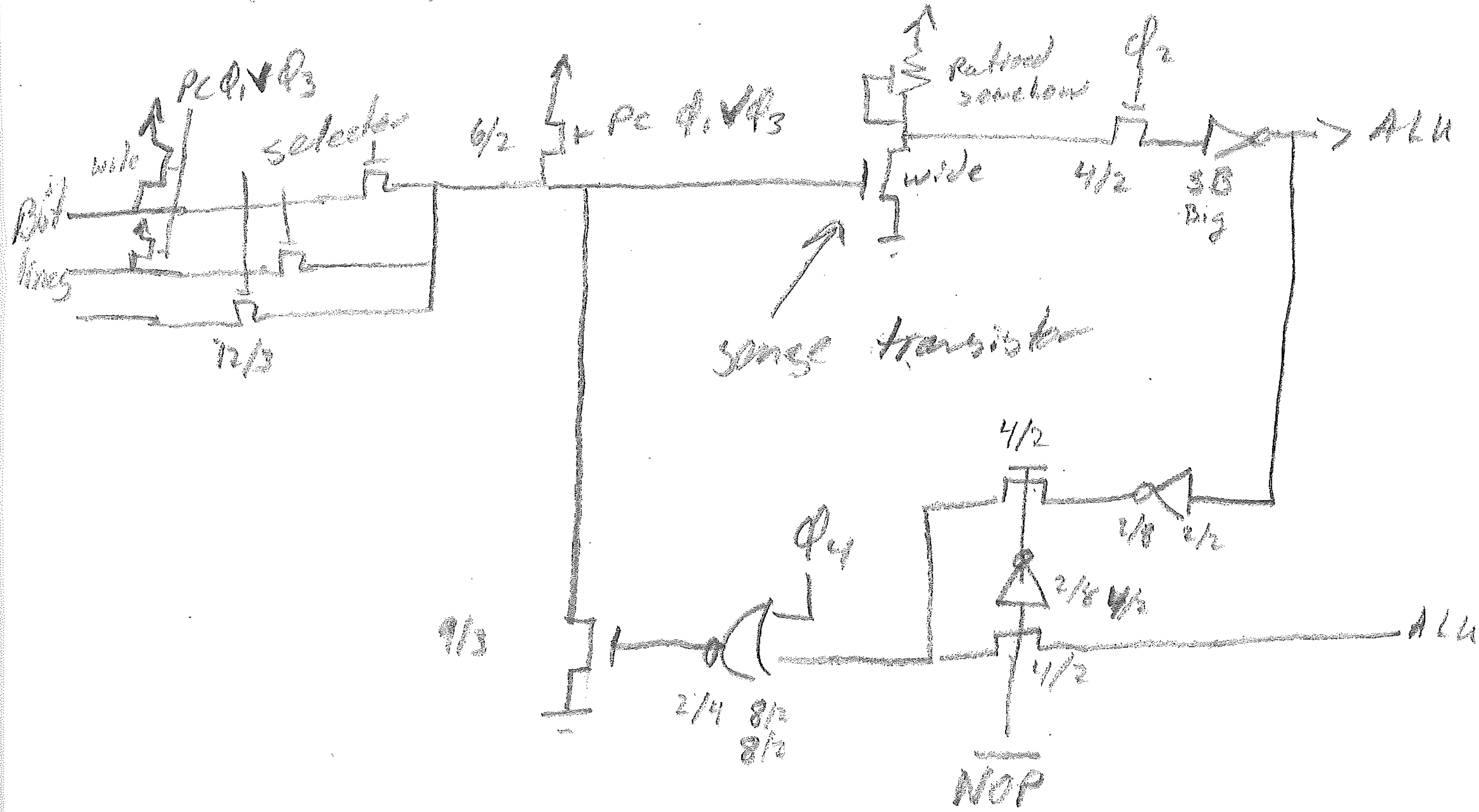
- Snap shot
- load into real register (POP)
- check parity → global (refresh) (crackle)
- ~~enable ring input (? or ring)~~
- pull ring (with ~~Sh~~ static shift reg)
- shift output daisy

Enter Instruction

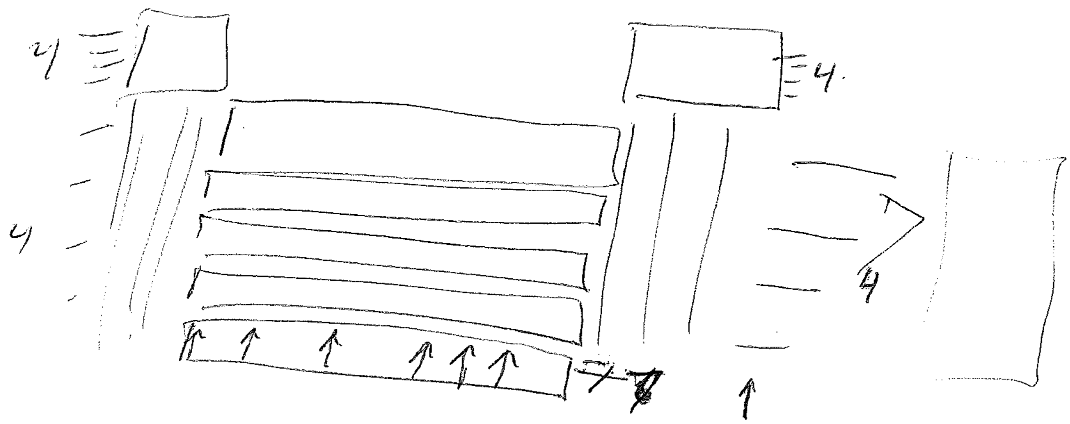
CTU



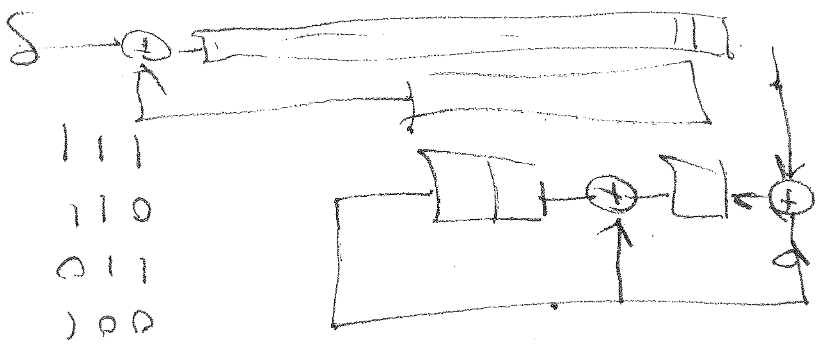
Safe RAM Shared Driver idea BLK 5/15/83



During Φ4 the bit line is pulled down or left high



170
4
680

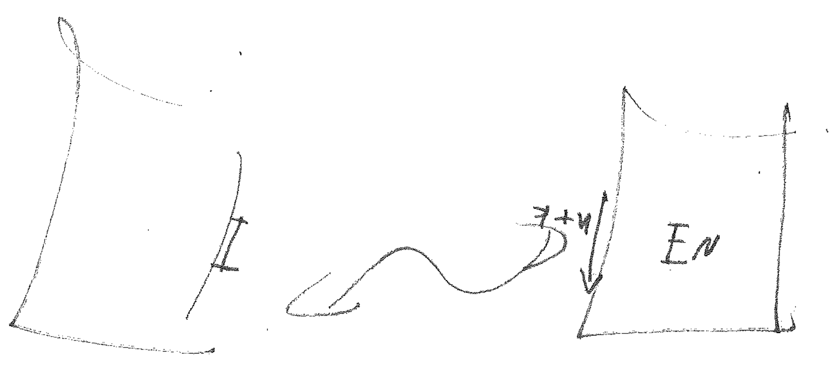


111
110
011
100
010
001

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

# of parity bits

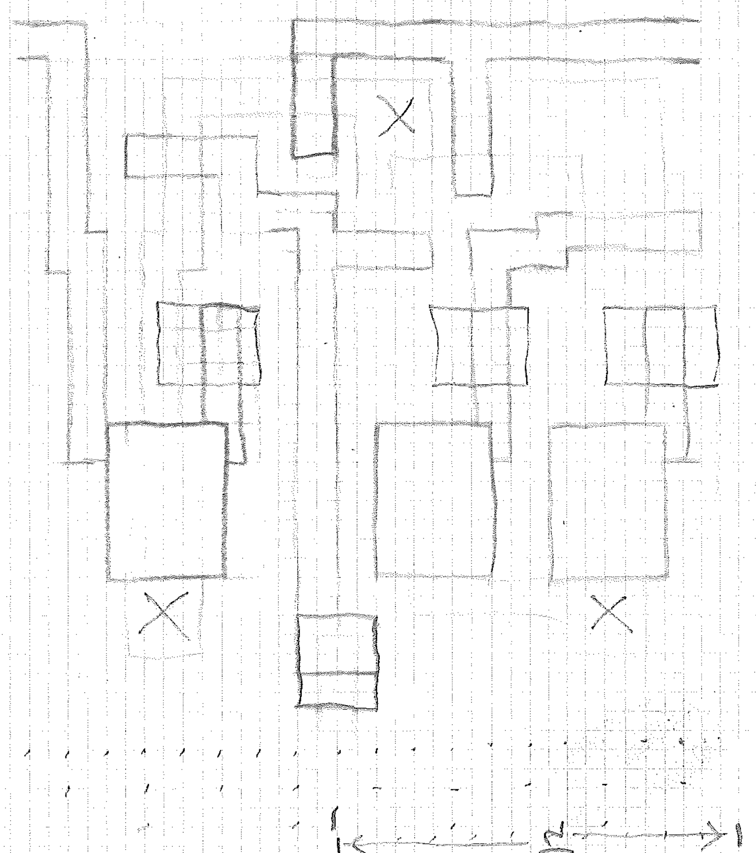
$2^{deg} - 1$ = total length



$\bar{h} + 8$

$\vec{9} + 06$

And



Daisy

Owen

also <kahle.spice>default.prm

1, recent,,
Return-path: <VLSI@USC-ISIF>
Date: 12 May 1983 8:15AM-PDT
From: The MOSIS System <VLSI at USC-ISIF> (send requests to MOSIS@ISIF)
To: kahle@mc
Subject: TOPIC: m2na.prm

*** EOOH ***
Date: Thursday, 12 May 1983 11:15-EDT
From: The MOSIS System <VLSI at USC-ISIF> (send requests to MOSIS@ISIF)
To: kahle at mc
Re: TOPIC: m2na.prm

;PS:<MOSIS.INFO>M2NA.PRM.2, 29-Jan-83 15:10:13, Ed: B-SMITH@USC-ISIF

M2NA ISI Accutest, HP/Sierracin, NMOS/MC2, 3 in, 3u, (Level = 2)

10 wafers probed, total number of dies: 420

EXECUTIVE SUMMARY

Tst	Count	Mean	Sigma	Sig/Mean		
34	419	0.791	0.01	1.36%	(V)	Vth Large Enh (Vbs = 0)
45	360	-3.109	0.66	-21.14%	(V)	Vth Large Dep (Vbs = 0)
11	419	4.158	0.54	13.05%	(um)	Metal Width Narrow (3 Lambda)
84	305	3.000	0.20	6.57%	(um)	Poly Width Narrow (2 Lambda)
87	418	2.558	0.14	5.45%	(um)	Diff Width Narrow (2 Lambda)
65	376	1.986	0.19	9.74%	(V)	Vinv (K8)
69	339	2.393	0.13	5.26%	(V)	Vinv (K4)
80	372	18.489	3.93	21.27%	(MHz)	Ring Freq 5.0V

COMPLETE TEST RESULTS

Tst	Count	Mean	Sigma	Sig/Mean		
1	339	-3.092	0.65	-20.98%	(V)	Vth Dep Length 10L Width 2L
2	417	0.395	0.11	27.12%	(ua/v)	Slope Dep Width 2L
3	332	-3.010	0.56	-18.51%	(V)	Vth Dep Width 4L
4	416	0.843	0.23	27.26%	(ua/v)	Slope Dep Width 4L
5	335	-3.559	0.91	-25.55%	(V)	Vth Dep Width 8L
6	417	1.503	0.48	31.66%	(ua/v)	Slope Dep Width 8L
7	341	-3.437	0.66	-19.10%	(V)	Vth Dep Length 2L Width 2L
8	417	1.633	0.47	28.86%	(ua/v)	Slope Dep Length 2L Width 2L
14	419	0.780	0.01	1.18%	(V)	Vth Enh Width 10L Length 2L
15	419	12.296	0.68	5.52%	(ua/v)	Slope Enh Length 2L
16	418	0.787	0.01	1.19%	(V)	Vth Enh Length 4L
17	418	6.436	0.18	2.83%	(ua/v)	Slope Enh Length 4L
18	419	0.796	0.01	1.40%	(V)	Vth Enh Length 8L
19	419	3.235	0.06	1.89%	(ua/v)	Slope Enh Length 8L
20	331	-3.623	0.71	-19.57%	(V)	Vth Dep Length 2L
21	419	8.063	2.47	30.64%	(ua/v)	Slope Dep Length 2L
22	338	-3.207	0.53	-16.46%	(V)	Vth Dep Length 4L
23	419	5.063	1.43	28.24%	(ua/v)	Slope Dep Length 4L
24	344	-3.058	0.59	-19.35%	(V)	Vth Dep Length 8L
25	417	2.885	0.81	28.20%	(ua/v)	Slope Dep Length 8L
26	418	0.847	0.01	1.68%	(V)	Vth Enh Length 10L Width 2L
27	416	0.461	0.02	4.36%	(ua/v)	Slope Enh Width 2L
28	418	0.812	0.01	1.06%	(V)	Vth Enh Width 4L
29	418	0.955	0.03	2.66%	(ua/v)	Slope Enh Width 4L
30	418	0.800	0.01	1.11%	(V)	Vth Enh Width 8L
31	418	1.982	0.04	1.78%	(ua/v)	Slope Enh Width 8L
32	417	0.822	0.01	1.38%	(V)	Vth Enh Length 2L Width 2L
33	417	2.198	0.16	7.31%	(ua/v)	Slope Enh Length 2L Width 2L
34	419	0.791	0.01	1.36%	(V)	Vth Large Enh (Vbs = 0)
35	419	2.620	0.03	1.18%	(ua/v)	Slope Large Enh (Vbs = 0)
36	419	52.404	0.62	1.18%	(ua/v**2)	Kp Large Enh (Vbs = 0)
37	419	1.142	0.02	1.41%	(V)	Vth Large Enh (Vbs = -1)
38	419	2.537	0.03	1.23%	(ua/v)	Slope Large Enh (Vbs = -1)
39	419	50.748	0.62	1.23%	(ua/v**2)	Kp Large Enh (Vbs = -1)
40	419	0.751	0.02	2.73%	(V**-0.5)	Gamma Lg Enh (Vbs = 0,-1)
41	419	1.460	0.02	1.41%	(V)	Vth Large Enh (Vbs = -5)
42	419	2.456	0.03	1.26%	(ua/v)	Slope Large Enh (Vbs = -5)
43	419	49.111	0.62	1.26%	(ua/v**2)	Kp Large Enh (Vbs = -5)
44	419	0.293	0.01	2.89%	(V**-0.5)	Gamma Lg Enh (Vbs = -1,-5)
45	360	-3.109	0.66	-21.14%	(V)	Vth Large Dep (Vbs = 0)
46	419	2.403	0.56	23.47%	(ua/v)	Slope Large Dep (Vbs = 0)
47	419	48.054	11.28	23.47%	(ua/v**2)	Kp Large Dep (Vbs = 0)
48	330	-2.772	0.23	-8.13%	(V)	Vth Large Dep (Vbs = -1)
49	411	2.359	0.62	26.26%	(ua/v)	Slope Large Dep (Vbs = -1)
50	411	47.171	12.39	26.26%	(ua/v**2)	Kp Large Dep (Vbs = -1)
51	325	0.296	0.13	44.43%	(V**-0.5)	Gamma Lg Dep (Vbs = 0,-1)
52	333	-2.520	0.21	-8.42%	(V)	Vth Large Dep (Vbs = -5)
53	413	2.358	0.62	26.45%	(ua/v)	Slope Large Dep (Vbs = -5)
54	413	47.159	12.48	26.45%	(ua/v**2)	Kp Large Dep (Vbs = -5)
55	370	0.292	0.22	73.99%	(V**-0.5)	Gamma Lg Dep (Vbs = -1,-5)
56	419	0.027	0.00	10.54%	(nA)	Idss Large Enh
57	420	106.379	7.26	6.82%	(pF)	Gate Capacitor (Vgate = 5V)
58	415	0.028	0.04	125.50%	(nA)	Gate (Cap) Leak
59	344	0.335	0.14	42.59%	(V)	Vout Z Pulsed (K12)
60	0	0.000	0.00	42.59%	(V)	Vinv Pulsed (K12)
61	417	1.889	0.39	20.54%	(V)	Vinv (K12)
62	375	-10.112	2.57	-25.45%		Gain at Vinv (K12)
63	331	0.496	0.12	24.79%	(V)	Vout Z Pulsed (K8)
64	0	0.000	0.00	24.79%	(V)	Vinv Pulsed (K8)
65	376	1.986	0.19	9.74%	(V)	Vinv (K8)
66	376	-7.759	2.02	-26.00%		Gain at Vinv (K8)
67	419	5.000	0.00	0.03%	(V)	Vout high (K4)

68	338	0.381	0.09	24.34%	(V)	Vout low (K4)
69	339	2.393	0.13	5.26%	(V)	Vinv (K4)
70	334	-4.795	0.74	-15.43%		Gain at Vinv (K4)
75	418	12.998	0.79	6.06%	(V)	V Metalfield Vbs = 0, Id = 1ua
76	419	19.718	0.70	3.53%	(V)	V Metalfield Vbs = -1, Id = 1ua
77	1	19.240	0.00	0.00%	(V)	V Polyfield Vbs = 0, Id = 1ua
78	419	12.158	0.36	2.99%	(V)	V Polyfield Vbs = -1, Id = 1ua
79	372	18.721	3.56	19.00%	(MHz)	Ring Freq 4.0V
80	372	18.489	3.93	21.27%	(MHz)	Ring Freq 5.0V
81	372	17.964	4.21	23.42%	(MHz)	Ring Freq 6.0V
9	418	33.590	1.29	3.85%	(mOhm/sq)	Metal Sheet Resisitance
10	419	7.213	3.56	49.29%	(um)	Metal Width Wide (3 Lambda + 3 um)
11	419	4.158	0.54	13.05%	(um)	Metal Width Narrow (3 Lambda)
82	305	30.680	3.21	10.45%	(Ohm/sq)	Polysheet Resistance
83	305	5.939	0.27	4.59%	(um)	Poly Width Wide (2 Lambda + 3 um)
84	305	3.000	0.20	6.57%	(um)	Poly Width Narrow (2 Lambda)
85	419	42.542	0.62	1.45%	(Ohm/sq)	Diffusion Resistance
86	418	5.293	0.14	2.60%	(um)	Diff Width Wide (2 Lambda + 3 um)
87	418	2.558	0.14	5.46%	(um)	Diff Width Narrow (2 Lambda)
88	360	8.533	0.59	6.94%	(Ohm)	M/D Contact Resistance (1Ma)
89	415	0.006	0.04	633.12%	(mV)	M/D Contact Vsum (1Ma)
90	365	8.469	0.59	7.00%	(Ohm)	M/D Contact Resistance (10Ma)
91	4	0.304	1.14	375.78%	(mV)	M/D Contact Vsum (10Ma)
12	420	3.133	0.22	6.88%	(Ohm)	Buried Contact Resistance
13	416	1.013	0.32	31.56%	(mV)	Buried Contact Vsum
71	419	5.242	0.73	13.90%	(Ohm)	M/P Contact Resistance (1Ma)
72	418	0.025	0.01	30.77%	(mV)	M/P Contact Vsum (1Ma)
73	419	5.197	0.73	13.97%	(Ohm)	M/P Contact Resistance (10Ma)
74	420	-0.772	0.16	-20.98%	(mV)	M/P Contact Vsum (10Ma)

End of run

^_

tems. Designs implemented according to these rules are easily scaled and may have reasonable longevity.

2.7 ELECTRICAL PARAMETERS

By satisfying the constraints imposed by the design rules, designers may create circuit layout patterns with the knowledge that the appropriate transistors, lines, etc., produced by the wafer-fabrication process will be as originally specified in their layout patterns. To complete a design it is necessary to also know the electrical parameters of the transistors, diffused layers, polysilicon layers, etc., so that the performance of circuits can be evaluated. The resistances per square of the various layers and the capacitance per square micron with respect to underlying substrate are shown in Table 2.1. Note that the resistance of a square of material contacted along two opposite sides is independent of the size of the square and equals the resistivity of the material divided by its thickness. The tabulated values are typical of processes running in 1978. As the circuit dimensions are scaled *down* by dividing by a factor α, the parameters scale approximately as described in the table.

The relative resistance values of metal, diffusion, poly, and drain-to-source paths of transistors are quite different. Diffused layers and good polysilicon layers have more than one hundred times the resistance per square area of the metal layer. A fully turned-on transistor has approximately one thousand times the resistance of the diffused and polysilicon layers. The capacitances are not as wildly different as the resistances of the various layers. Compare the capacitances in Table 2.1 to the gate-to-channel capacitance, as a reference. The diffused areas typically have one fourth the capacitance per square micron. Polysilicon on thick oxide has approximately one tenth, and the metal layer slightly less than one tenth, of the gate-channel capacitance per square micron.

ps; ss <9 lr> design notes

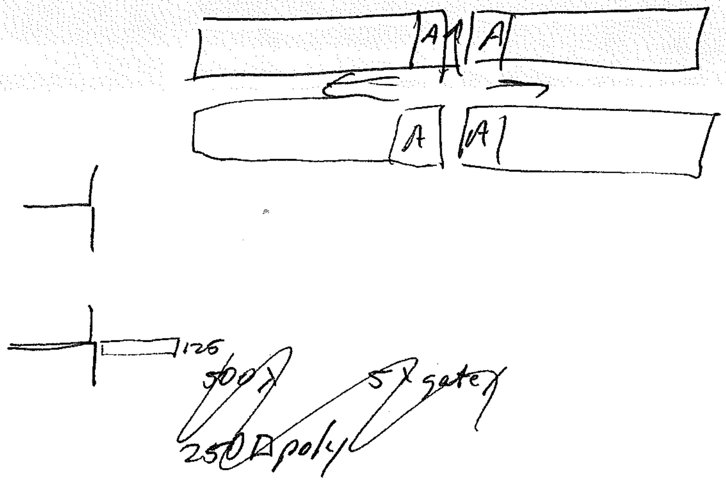
The relative values of the resistances and capacitances are not expected to vary dramatically as the processes evolve toward smaller dimensions, with the exception of the transistor resistance per square, which is independent of α.

Table 2.1 Typical MOS electrical parameters (1978).

Resistances		
Metal	≈ 0.03 ohms/□	(Resistances/square scale *up* by α,
Diffusion	≈ 10 ohms/□	as dimensions scale *down* by α,
Poly	≈ 15-100 ohms/□	except that the transistor R/□
Transistor	≈ 10^4 ohms/□	is independent of α.)
Capacitances		9 × 10^{-4} pF/μm sidewall
Gate-channel	≈ 4 × 10^{-4} pf/μm²	(Capacitances/micron² scale
Diffusion	≈ 1 × 10^{-4} pf/μm²	*up* by α, as dimensions
Poly	≈ 0.4 × 10^{-4} pf/μm²	scale *down* by α.)
Metal	≈ 0.3 × 10^{-4} pf/μm²	poly sidewall 5 × 10^{-4} pF/μm

F/m² = 10^{12} pF/ $(10^6 \mu m)^2$

buried contact


A
A
A
A
125
500
5 X gate
250 poly